1/37

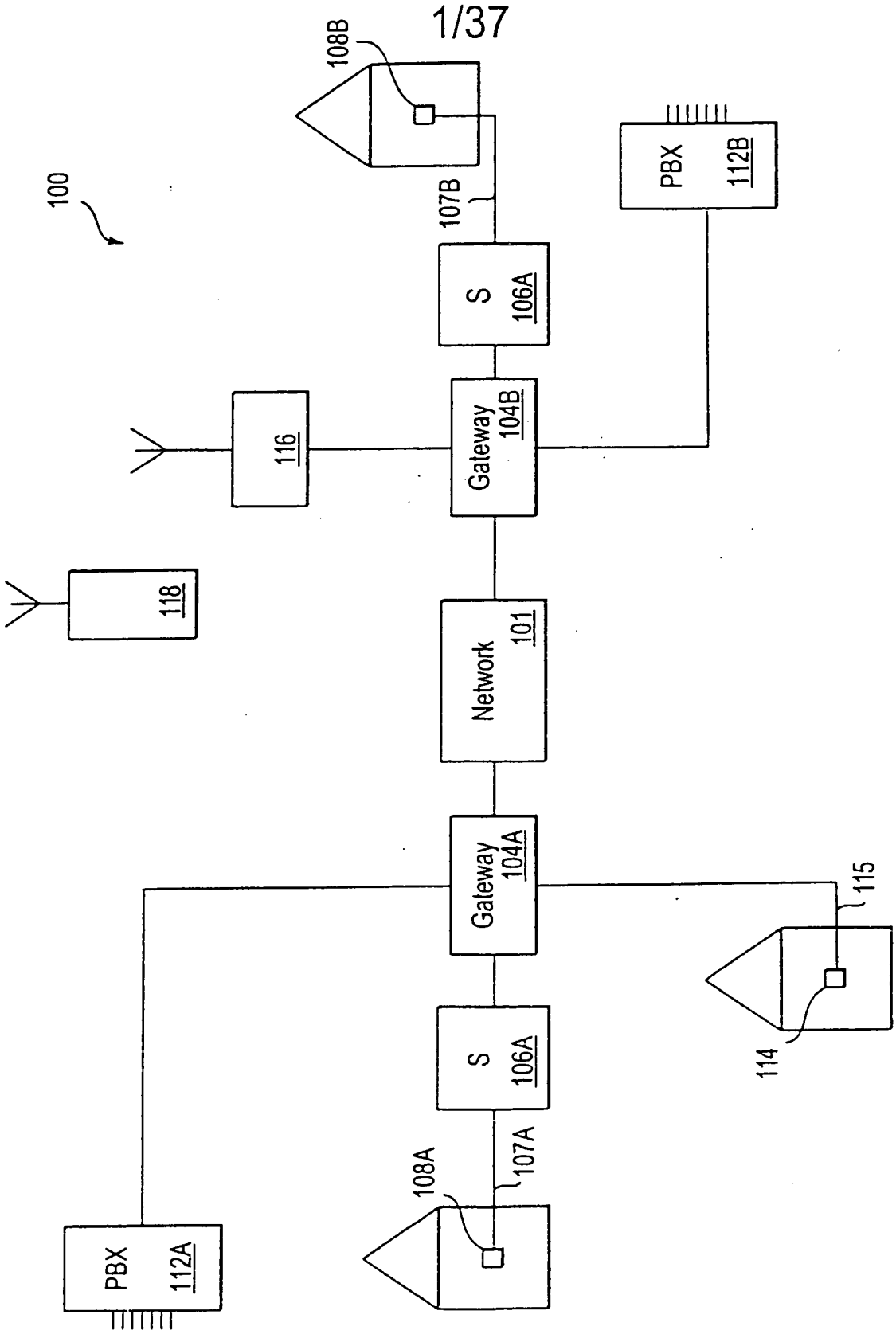

100
108B
107B
S 106A
PBX 112B
116
Gateway 104B
118
Network 101
Gateway 104A
115
114
S 106A
107A
108A
PBX 112A

Fig. 1A

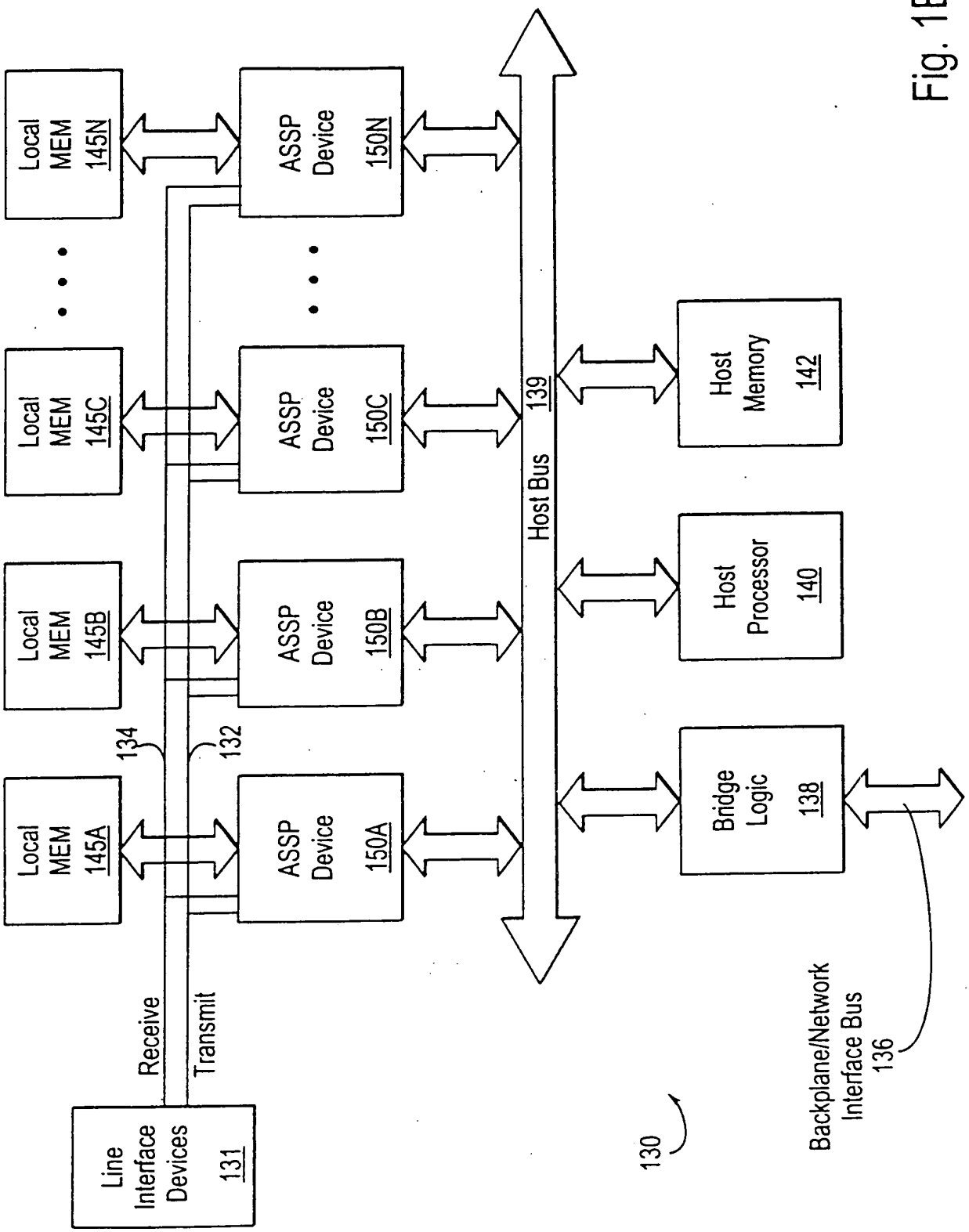

Local MEM 145N
ASSP Device 150N
Local MEM 145C
ASSP Device 150C
Local MEM 145B
ASSP Device 150B
Local MEM 145A
ASSP Device 150A
Host Bus 139
Host Memory 142
Host Processor 140
Bridge Logic 138
Backplane/Network Interface Bus 136
134
132
Receive
Transmit
Line Interface Devices 131
130

Fig. 1B

3/37

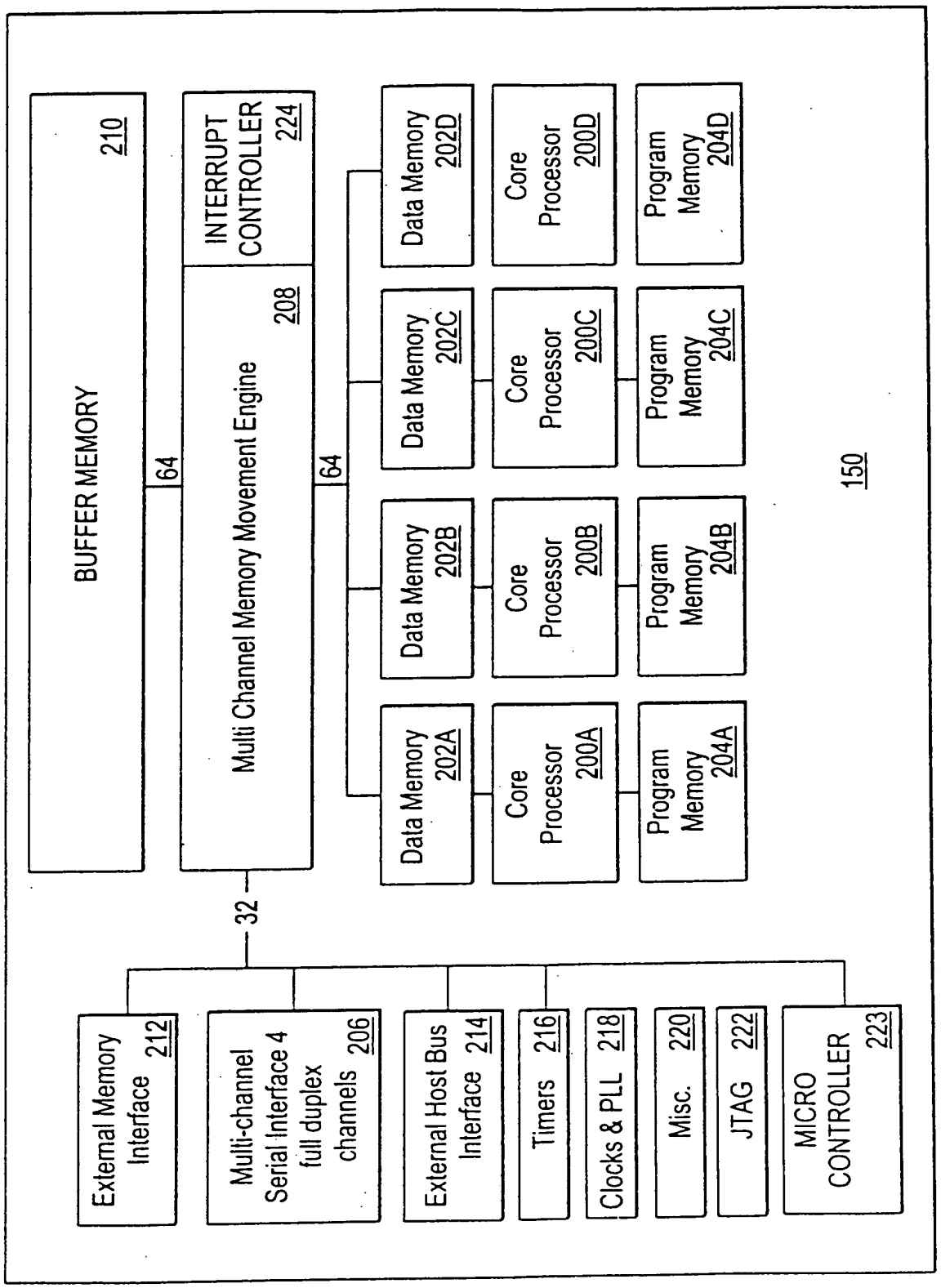

BUFFER MEMORY 210
INTERRUPT CONTROLLER 224
Multi Channel Memory Movement Engine 208
64
64
32
Data Memory 202A
Core Processor 200A
Program Memory 204A
Data Memory 202B
Core Processor 200B
Program Memory 204B
Data Memory 202C
Core Processor 200C
Program Memory 204C
Data Memory 202D
Core Processor 200D
Program Memory 204D
150
External Memory Interface 212
Multi-channel Serial Interface 4 full duplex channels 206
External Host Bus Interface 214
Timers 216
Clocks & PLL 218
Misc. 220
JTAG 222
MICRO CONTROLLER 223

Fig. 2

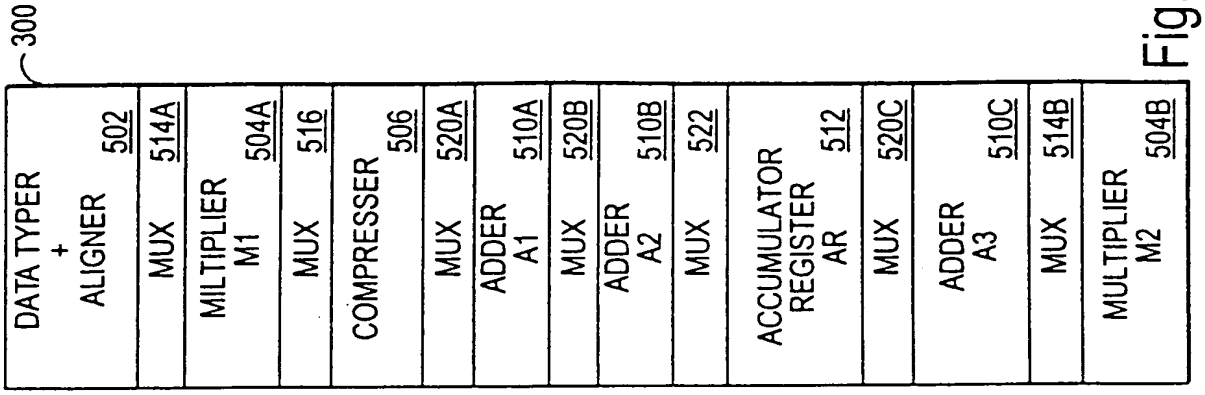


Fig. 5A

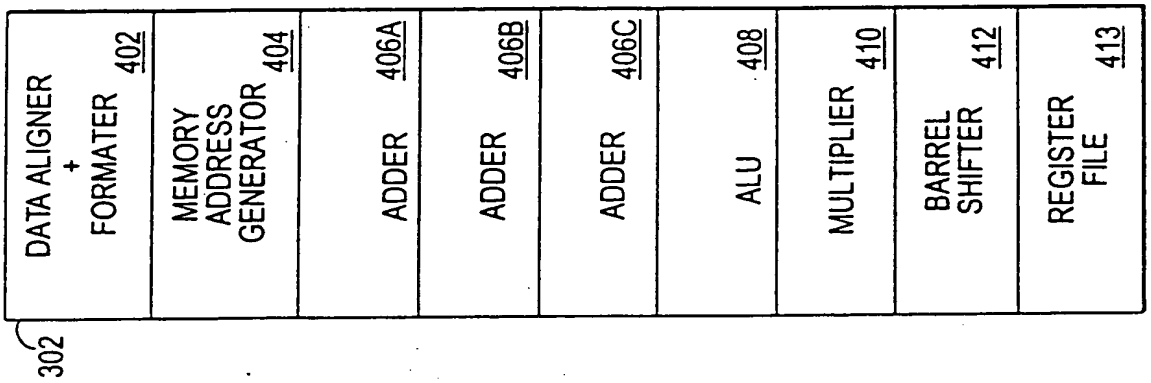


Fig. 4

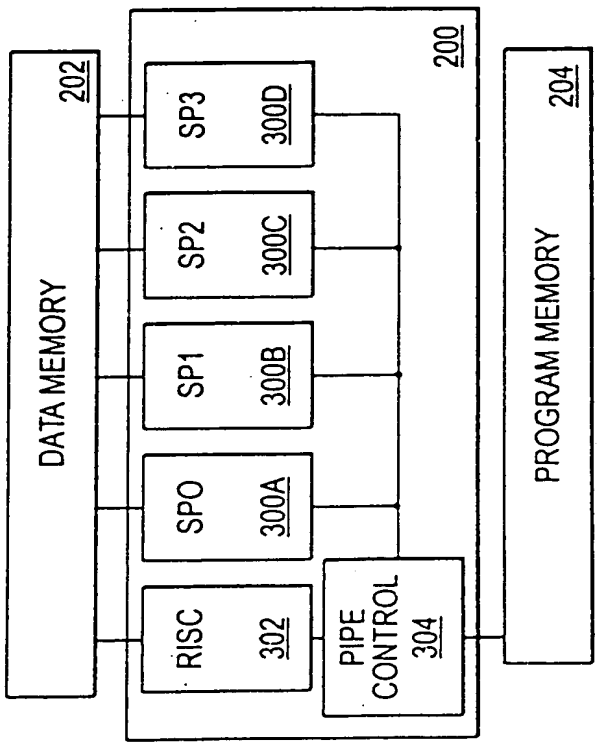


Fig. 3

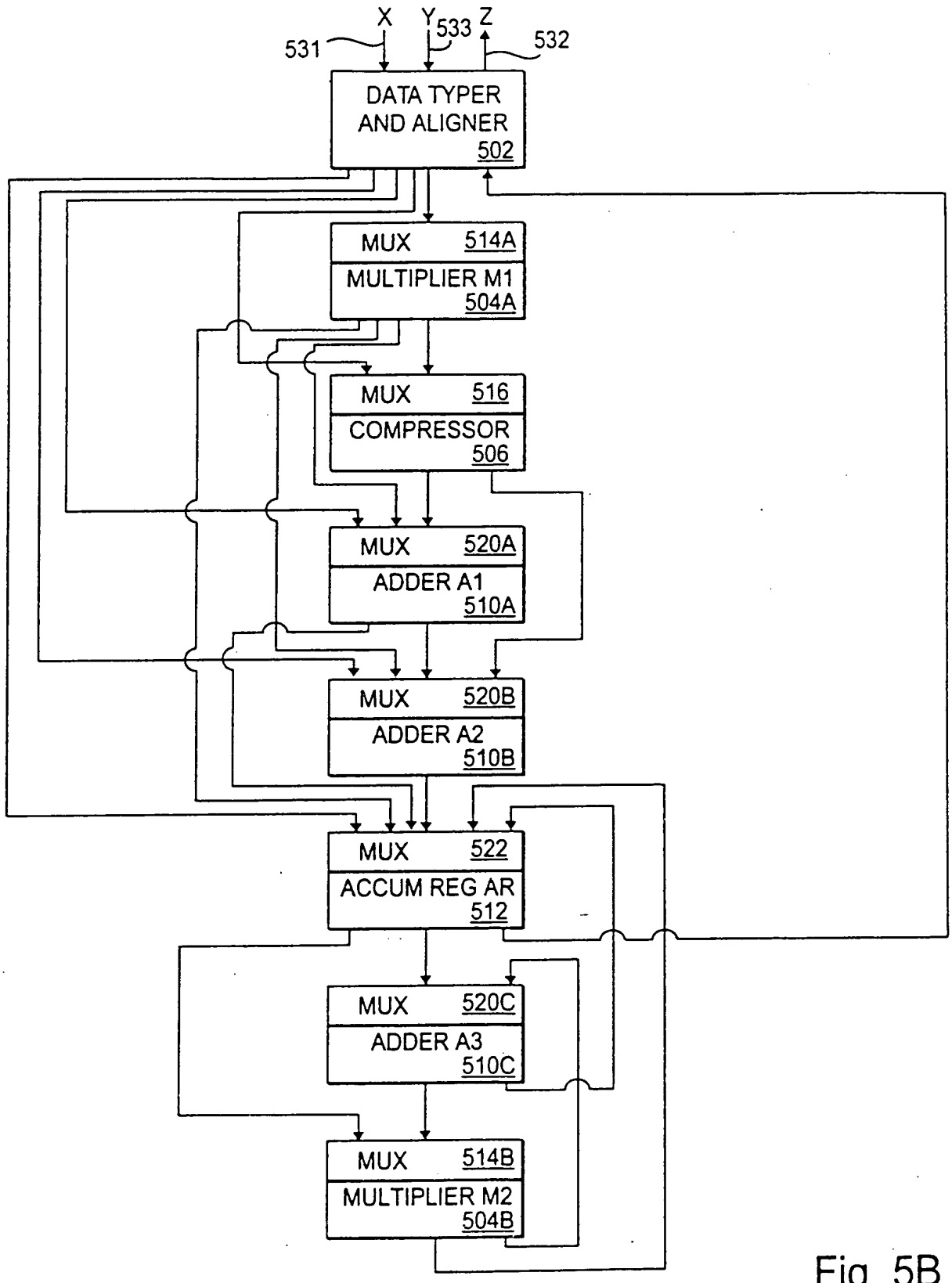

X
Y
Z
531
533
532
DATA TYPER AND ALIGNER 502
MUX 514A
MULTIPLIER M1 504A
MUX 516
COMPRESSOR 506
MUX 520A
ADDER A1 510A
MUX 520B
ADDER A2 510B
MUX 522
ACCUM REG AR 512
MUX 520C
ADDER A3 510C
MUX 514B
MULTIPLIER M2 504B

Fig. 5B

6/37

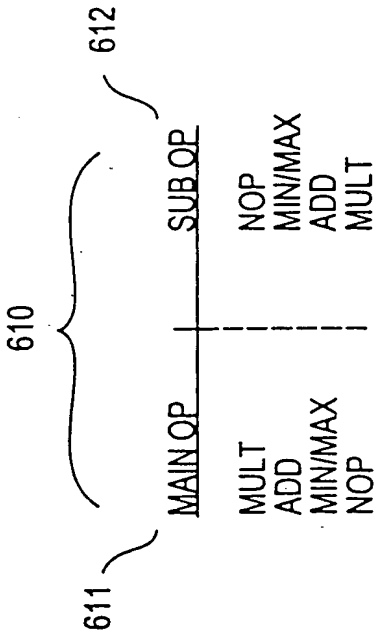


Fig. 6B

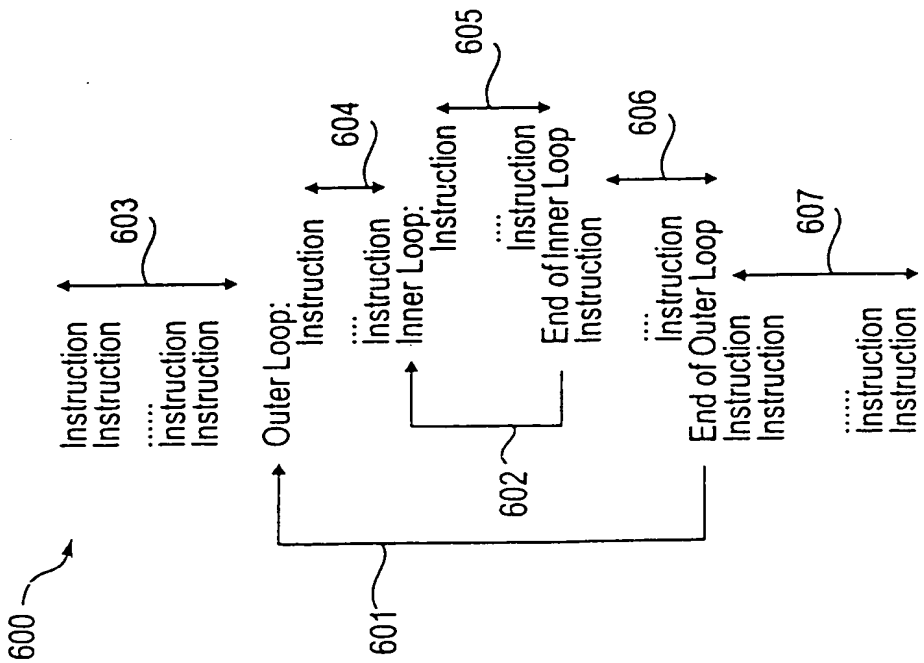


Fig. 6A

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

1	0	0	PS	S*	SX	SX	V/S	SA	DA	Sub-op			1	Pred	PL	Sxt	Syt	Rnd	S*	S*	S*	0	SA	DA	abs	0	0

da = +/-sx*sy	Nop	0	0	0
da = +/-(sx*sy) + sa	Add	0	0	1
da = +/-(sx*sa) + sy	Add	0	1	0
da = +/-(sx*sy) - sa	Sub	0	1	1
da = +/-(sx*sa) - sy	Sub	1	0	0
da = min(+/-sx*sy,sa)	Min	1	0	1
da = min(+/-sx*sa,sy)	Min	1	1	0
da = max(+/-sx*sy,sa)	Max	1	1	1

Lt
Lt
Lt
Lt
Gx
Gx
Gx

Fig. 6C

7/37

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20		
1	0	0	PS	S*	SX					SY				V/S	SA	DA	0	1	0	Add	da = +/-(mx*sa) + my
																	1	0	0	Sub	da = +/-(mx*sa) - my
																	1	1	0	Min	da = min(+/-mx*sa,my)

Fig. 6D

8/37

20-bit ISA

39	19		
0	0	20-bit parallel	Control II Control
0	1	20-bit serial	Control # Control
1	0	40-bit extended	DSP extensions/Shadow
1	1	20-bit serial	DSP # DSP

DSP Instructions

39	38	37	36	35	35	33	32	31	30	29	28	27	26	25	24	23	22	21	20	
Multipy	1	0	0	PS		S*	SX				SY			V/S	SA	DA	Sub-op			
							da = sx*sy										0	0	0	Nop
							da = (sx*sy) + sa										0	0	1	Add
							da = (sx*sa)+ sy										0	1	0	Add
							da = (sx*sy) - sa										0	1	1	Sub
							da = (sx*sa) - sy										1	0	0	Sub
							da = min(sx*sy,sa)										1	0	1	Min
							da = min(sx*sa,sy)										1	1	0	Min
							da = min(sx*sy,sa)										1	1	1	Max
Add	1	0	1	PS		+/-	SX				SY			V/S	SA	DA	Sub-op			
							da = sx+sy										0	0	0	Nop
							da = sx+sy+sa										0	0	1	Add
							da = sx+sy;sa=sx-sy;										0	1	0	AddSub
							da = (sx+ sy)*sa										0	1	1	Mul
							da = -(sx+sy)*sa										1	0	0	MulN
							da = min(sx+sy,sa)										1	0	1	Min
							da = max(sx+sy,sa)										1	1	0	Max
							da = ssum(sa) (sx,sy unused)										1	1	1	CombAdd
Extremum	1	1	0	PS		X/N	SX				SY			V/S	SA	DA	Sub-op			
							da = ext(sx,sy)										0	0	0	Nop
							da = ext(sx,sy,sa)										0	0	1	Ext
							da = ext(sx,sa) *sy										0	1	0	Mul
							da = -ext(sx,sa) *sy										0	1	1	MulN
							da = ext(sx,sa) + sy										1	0	0	Add
							da = ext(sx,sa) - sy										1	0	1	Sub
							ext(sa,da)?t = sx,tr = sy,lcs = lc										1	1	0	amax
type-match	1	1	0	PS		0	SX				SY			x	x	x	1	1	1	
Permute	1	1	0	PS		1	SX				Type			x	ereg		1	1	1	Permute
Reserved	1	1	1	PS		x	SX				SY			SA	DA	V/S	Sub-op			

Fig. 6E(1)

9/37

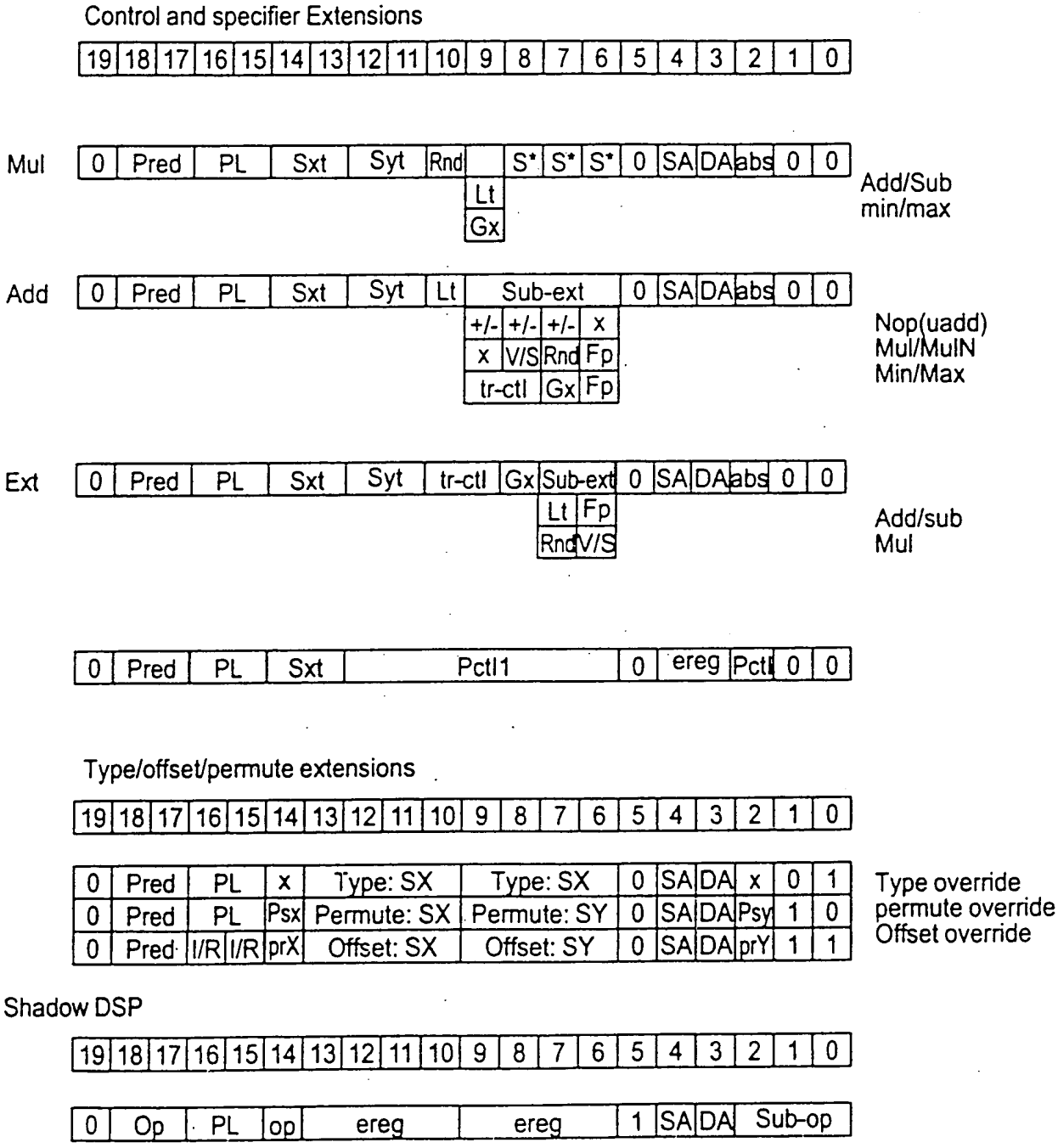

Control and specifier Extensions
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Mul
0 Pred PL Sxt Syt Rnd S* S* S* 0 SA DA abs 0 0
Lt
Gx
Add/Sub
min/max
Add
0 Pred PL Sxt Syt Lt Sub-ext 0 SA DA abs 0 0
+/- +/- +/- x
x V/S Rnd Fp
tr-ctl Gx Fp
Nop(uadd)
Mul/MulN
Min/Max
Ext
0 Pred PL Sxt Syt tr-ctl Gx Sub-ext 0 SA DA abs 0 0
Lt Fp
Rnd V/S
Add/sub
Mul
0 Pred PL Sxt Pctl1 0 ereg Pctl 0 0
Type/offset/permute extensions
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 Pred PL x Type: SX Type: SX 0 SA DA x 0 1
0 Pred PL Psx Permute: SX Permute: SY 0 SA DA Psy 1 0
0 Pred I/R I/R prX Offset: SX Offset: SY 0 SA DA prY 1 1
Type override
permute override
Offset override
Shadow DSP
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 Op PL op ereg ereg 1 SA DA Sub-op

Fig. 6E(2)

10/37

Control Instructions

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
add.sub	L	Pred		0	0	0	RX				RY				RZ				+/-	0	
max.min	L	Pred		0	0	0	RX				RY				RZ				X/N	1	
Shift	L	Pred		0	0	1	RX				UI4				RZ				UI1	R/L	<Bit1, Bits9-6> ==UI5 (Shift Amount)
Logic	L	Pred		0	1	0	RX				RY				RZ				&,	&I	
Mux	L	Pred		0	1	1	RX				RY				RZ				Pd	0	
mov	L	Pred		0	1	1	Rx				DZ				Rxt	Dzt	0	0	0	1	
addi	L	Pred		0	1	1	SI4				DZ				x	x	1	0	0	1	
mov2erg	L	Pred		0	1	1	RX				unit		ereg		gd	type		1	0	1	
Ldm	L	Pred		0	1	1	RX				Dz1				Dz2				1	1	
Set4bits	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI4				0	
Set2bits	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI2		0	0	1	
Setbit	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI1	UI1	1	0	0	<Bit3, Bits13-10> ==UI5: POS
Movl	L	Pred		1	0	0	SI8								RZ				1	1	
Jmp	L	Pred		1	0	1	SI9									0	Pred		0	0	
Call	L	Pred		1	0	1	SI9									1	Pred		0	0	
Loop	L	Pred		1	0	1	UI5: Lcount					UI5: Lsize						UI2:Lst	0	1	
Jmpi	L	Pred		1	0	1	RX				x	x	x	x	x	0	Pred		1	0	
Calli	L	Pred		1	0	1	RX				x	x	x	x	x	1	Pred		1	0	
Loopi	L	Pred		1	0	1	RX				x	UI5: Lsize						UI2:Lst	1	1	
Test	L	Pred		1	1	0	RX				RY				PZ			=,<,>		0	
Testbit	L	Pred		1	1	0	RX				UI5					PZ		B	0	1	
Andp.orp	L	Pred		1	1	0	Pa				Pb			Pc		PZ		&I	1	1	
Load	L	Pred		1	1	1	MX				RZ				Ext			0	0	0	
Store	L	Pred		1	1	1	MZ				RX				Ext			1	0	0	
eLoad	L	Pred		1	1	1	MX				RZ				1	1	1	0	0	0	
eStore	L	Pred		1	1	1	MZ				RX				1	1	1	1	0	0	
Extended	L	Pred		1	1	1	Blts 27:16												1	0	
Logic2	L	Pred		1	1	1	RX				RY/RZ				Rxt	Ryt	&,I,&I,I		0	1	
mov-erg	L	Pred		1	1	1	unit		ereg		RZ				gd	Sft		0	1	1	
Crb	L	Pred		1	1	1	RX				RZ				s/m	0	0	1	1	1	
Parity	L	Pred		1	1	1	RX				PZ			O/E	0	1	0	1	1	1	
Stm	L	Pred		1	1	1	MZ				RX				1	1	0	1	1	1	
Abs	L	Pred		1	1	1	RX				RZ				0	0	1	1	1	1	
Neg	L	Pred		1	1	1	RX				RZ				0	1	1	1	1	1	
Div-step	L	Pred		1	1	1	RX				RZ				1	0	1	1	1	1	
Test & Set	L	Pred		1	1	1	RX				PZ			0	1	1	1	1	1	1	
Return	L	Pred		1	1	1	Pred		I-ctl		0	1	0	1	1	1	1	1	1	1	
Zero-ac	L	Pred		1	1	1	ac#				1	1	0	1	1	1	1	1	1	1	
eSync	L	Pred		1	1	1	RZ				0	1	1	1	1	1	1	1	1	1	
Swi	L	Pred		1	1	1	UI3			0	1	1	1	1	1	1	1	1	1	1	
Nop	L	Pred		1	1	1	UI3			1	1	1	1	1	1	1	1	1	1	1	

Fig. 6F

11/37

Extended Control

	Bits 13:2 of upper half 39:20)															
	13	12	11	10	9	8	7	6	5	4	3	2	19	18	17	16
Insert/EXTRACT	RX				RZ				0	0	0	0	0	x	x	0
Inserti	UI4: length				RZ				0	0	0	1	0	x	x	0
Shift	RX				RZ				0	0	0	0	0	rxh	rxl	0
Rotate	RX				RZ				0	0	0	0	0	x	x	0
jmp. call	u17							J/C	0	0	1	0	0	Pred		0
dloop	U14: outer LC				U14: outer LC				0	0	1	1	0	x	exit	0
dloopi	RX				RY				0	0	1	1	0	x	exit	0
mult	RX				RY				0	1	0	0	0	x	x	0
add/sub	RX				RY				0	1	0	0	0	x	x	0
logicp	PX				D	PZ			0	1	0	0	0	x	x	0
Testi	RX				D	PZ			0	1	0	1	0	=,>,<		0
Movi	H/L	Fill			RZ				0	1	1	0	0	x	x	0
loadi	Type				RZ				0	1	1	1	0	x	x	0
storei	Type				RZ				0	1	1	1	0	x	x	0
loadt	RX				RZ				0	1	1	1	0	x	x	0
storet	MZ				RX				0	1	1	1	0	x	x	0
Add/subi	RX				RZ				1	0	+/-	0	0	LI	s/u	0
mini.maxi	RX				RZ				1	0	X/N	1	0	x	x	0
andi, ori	RX				RZ				1	1	&I	H/L	0	x	x	0

Fig. 6G(1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rxt	Rzt	I/E	R/I	R/I	Offset: UI5					Length: UI5					0
					x	RY				RV				x	
Rzt	UI5: Position				Imm10										
rzh	rzl	D	U/S	1	Shift: UI5					A/L	Lt	R/L	0	Fill	1
				0	ryh	RY									
x	x	x	x	1	Shift: UI15					1	1	R/L	1	x	1
				0	ryh	RY									

Fill: Sign/Zero

x	UI15														
UI1	UI4: outer L size			UI4: Inner L size				U12: 0-Ls		UI4 ; Inner L start					0
x	UI4: outer L size			UI4: Inner L size				U12: 0-Ls		UI4 ; Inner L start					1
0	rxh	rnd	ryh	+/-	=/+	RZ				I/f	rzh	rzl	s/u	s/u	0
0	rxh	rx1	ryh	ryl	+/-	RZ				Lt	rzh	rzl	x	x	1

BIT 15 is Continuation of Inner LC

1	T/F	T/F	T/F	&I	&I	PY		PV			x	1
Imm 16												
Imm 16												
0	0	Imm 14										
0	1	Imm 14										
1	Rzt	0	Type			S10						
1	Rzt	1	Type			S10						
Imm 16												
Imm 16												
Imm 16												

andp, orp, andorp, orandp: pz = (px relop py) relop pv)

Fig. 6G(2)

13/37

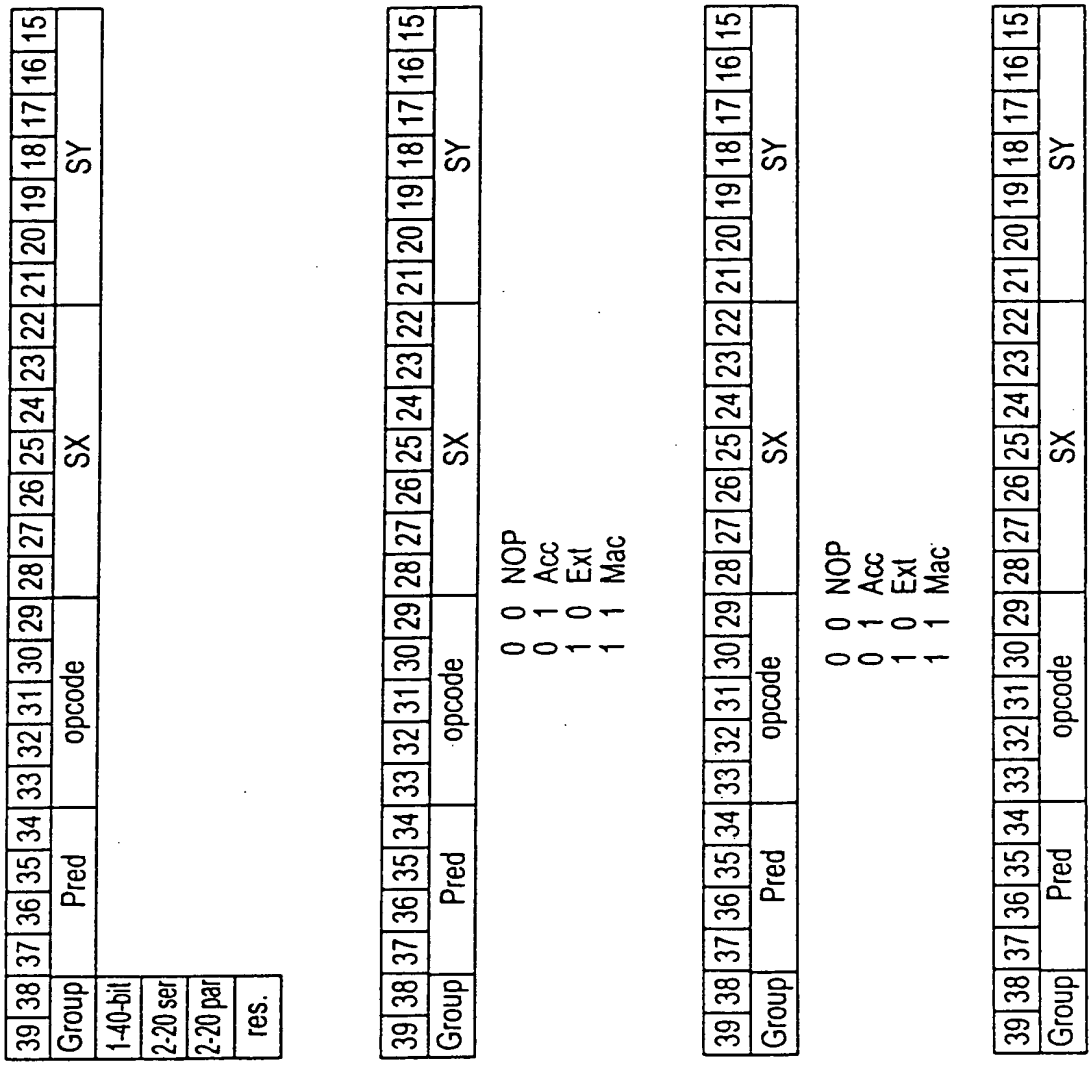

MAC:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15
Group
Pred
opcode
SX
SY
1-40-bit
2-20 ser
2-20 par
res.
ARITH:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15
Group
Pred
opcode
SX
SY
0 0 NOP
0 1 Acc
1 0 Ext
1 1 Mac
EXT:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15
Group
Pred
opcode
SX
SY
0 0 NOP
0 1 Acc
1 0 Ext
1 1 Mac
LOGIC:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15
Group
Pred
opcode
SX
SY

Fig. 6H(1)

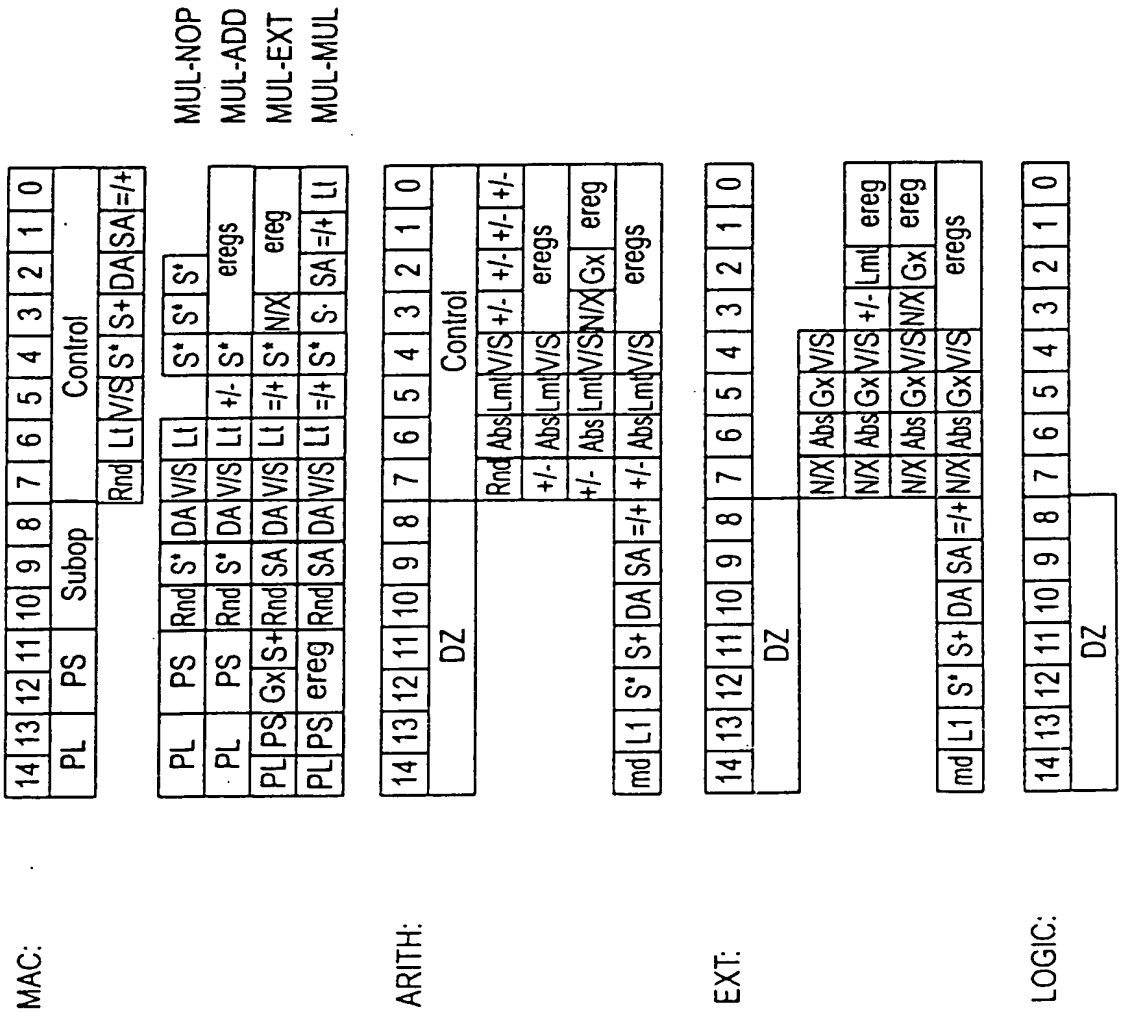

MAC:
MUL-NOP
MUL-ADD
MUL-EXT
MUL-MUL
ARITH:
EXT:
LOGIC:
14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PL PS Subop Control
Rnd LI V/S S* S+ DA SA =/+
PL PS Rnd S' DA V/S LI S* S' S' eregs
PL PS Rnd S' DA V/S LI +/- S* N/X ereg
PL PS Gx S+ Rnd SA DA V/S LI =/+ S* S- SA =/+ LI
PL PS ereg Rnd SA DA V/S LI =/+ S* S- SA =/+ LI
DZ Control
Rnd Abs Lmt V/S +/- +/- +/- +/-
+/- Abs Lmt V/S eregs
+/- Abs Lmt V/S N/X Gx ereg
md L1 S* S+ DA SA =/+ +/- Abs Lmt V/S eregs
DZ
N/X Abs Gx V/S
N/X Abs Gx V/S +/- Lmt ereg
N/X Abs Gx V/S N/X Gx ereg
md L1 S* S+ DA SA =/+ N/X Abs Gx V/S eregs
DZ

Fig. 6H(2)

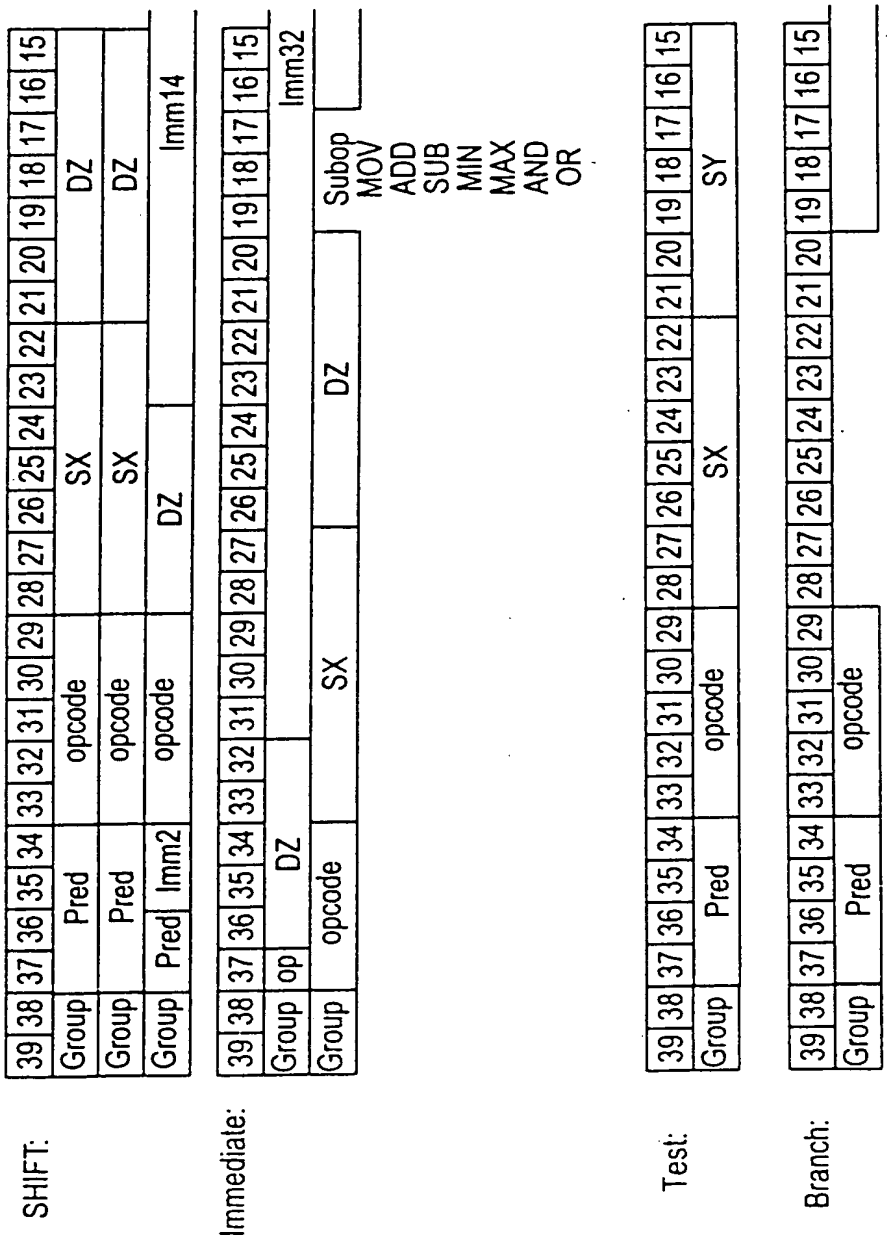

SHIFT:
Group Pred opcode SX DZ
Group Pred opcode SX DZ
Group Pred Imm2 opcode DZ Imm14
Immediate:
Group op DZ opcode SX DZ Subop Imm32
MOV
ADD
SUB
MIN
MAX
AND
OR
Test:
Group Pred opcode SX SY
Branch:
Group Pred opcode
Misc:

Fig. 6H(3)

16/37

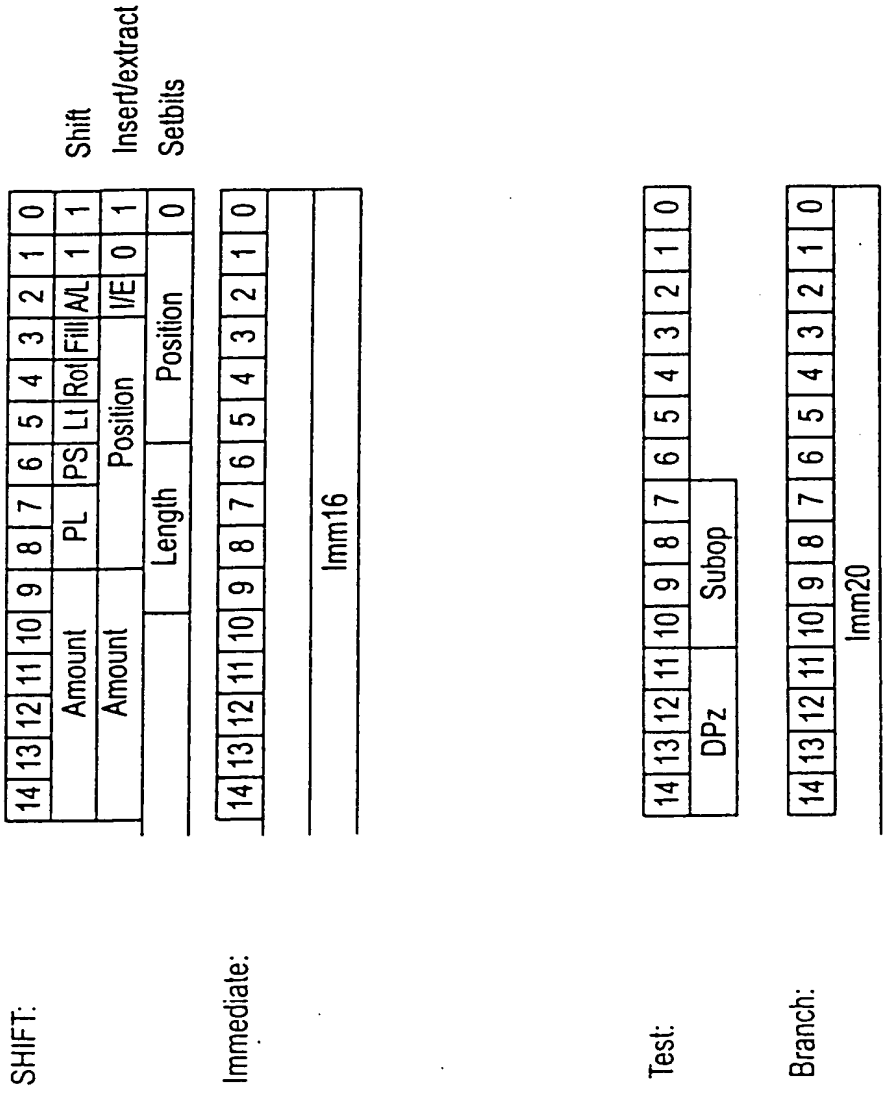

SHIFT:
Shift
Insert/extract
Setbits
14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Amount
PL
PS
Lt
Rot
Fill
A/L
Amount
Position
I/E
Length
Position
Immediate:
Imm16
Test:
DPz
Subop
Branch:
Imm20

Fig. 6H(4)

7-bit specifier: Parallel Store. Parallel Load in DSP Instructions

6	5	4	3	2	1	0

M/R

0	0	0	SPR: s0-s15	
0	0	1	reserved	
0	1	0	ac-names	
0	1	1	gpr:r0-r15	
1	0	ptr :(r0) to (r15)		off
1	1	onset: U14		ptr

Mem(ptr) ll ptr += ldr Always postupdate

Mem(ptr + ldr) ptr: p14, p15 Always preupdate

6-bit specifier: DSP Instructions

5	4	3	2	1	0

M/R

0	0	ac-names	
0	1	gpr: r0-r15	
1	ptr: r(0) to r(15)		off

Always postupdate

5-bit specifier: RISC Instructions

4	3	2	1	0

0	spr: s0-s15
1	gpr: r0-r15

4-bit specifier:

3	2	1	0

gpr: r0-r15		RISC Instructions
ptr: (r0-r7)	off	20-bit DSP Instructions
ereg		20-bit Shadow DSP Instructions

AR:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
type				permute					CB	Idx1:U13(0-7			xhr0: S15 (-16 to 15)					ptr															

Fig. 6i(1)

17/37

Blakely, Sokoloff, Taylor & Zafman LLP
Title: TONE DETECTION FOR INTEGRATED TELECOMMUNICATIONS PROCESSING
1st Named Inventor: Raghavendra S. Prabhu
Application No.: 09/938,699 Docket No.: 42390P12533
Sheet: 17 of 37
(714) 55[redacted]00

18/37

ac-names:

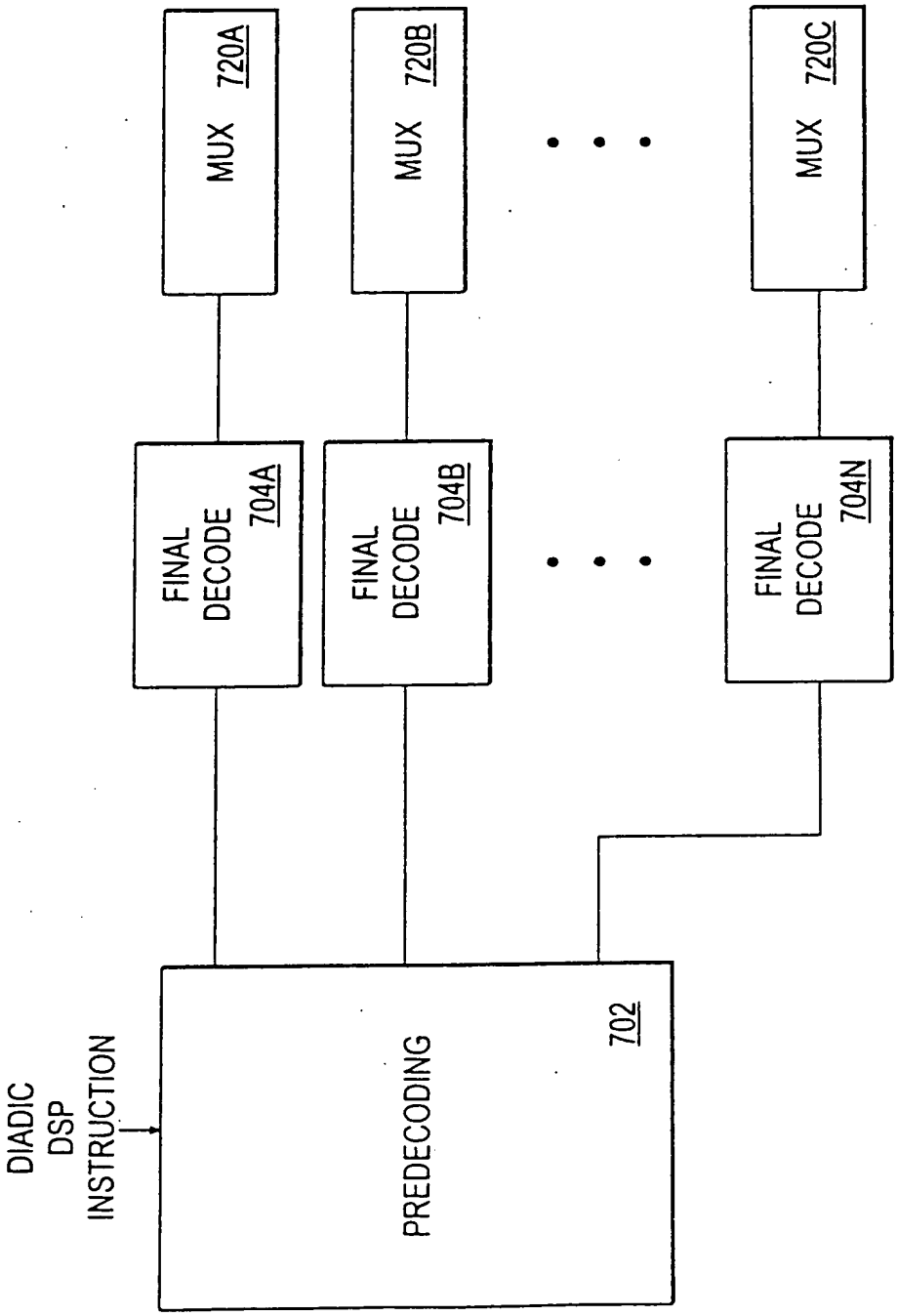
3	2	1	0		SPR:
0	0	0	0	AO (use type, SIMD)	gpr-type
0	0	0	1	A1	ereg-type
0	0	1	0	T	fu · ctl
0	0	1	1	TR	pls- ctf
0	1	0	0	A00 (unit 0)	cb - ctl
0	1	0	1	AI0	loop - ctl
0	1	1	0	T0	per
0	1	1	1	TRO	status
1	0	0	0	Sx1	
1	0	0	1	Sx1s	
1	0	1	0	Sx2	
1	0	1	1	Sx2s	
1	1	0	0	Sy1	
1	1	0	1	Sy1s	
1	1	1	0	Sy2	
1	1	1	1	Sy2s	

ereg-names

3	2	1	0	
0	0	0	0	AO
0	0	0	1	A1
0	0	1	0	T
0	0	1	1	TR
0	1	0	0	PP0
0	1	0	1	Aout
0	1	1	0	PP1
0	1	1	1	Dout
1	0	0	0	Sx1
1	0	0	1	Sx1s
1	0	1	0	Sx2
1	0	1	1	Sx2s
1	1	0	0	Sy1
1	1	0	1	Sy1s
1	1	1	0	Sy2
1	1	1	1	Sy2s

Fig. 6i(2)

Fig. 7


MUX 720A
MUX 720B
MUX 720C
FINAL DECODE 704A
FINAL DECODE 704B
FINAL DECODE 704N
PREDECODING 702
DIADIC DSP INSTRUCTION

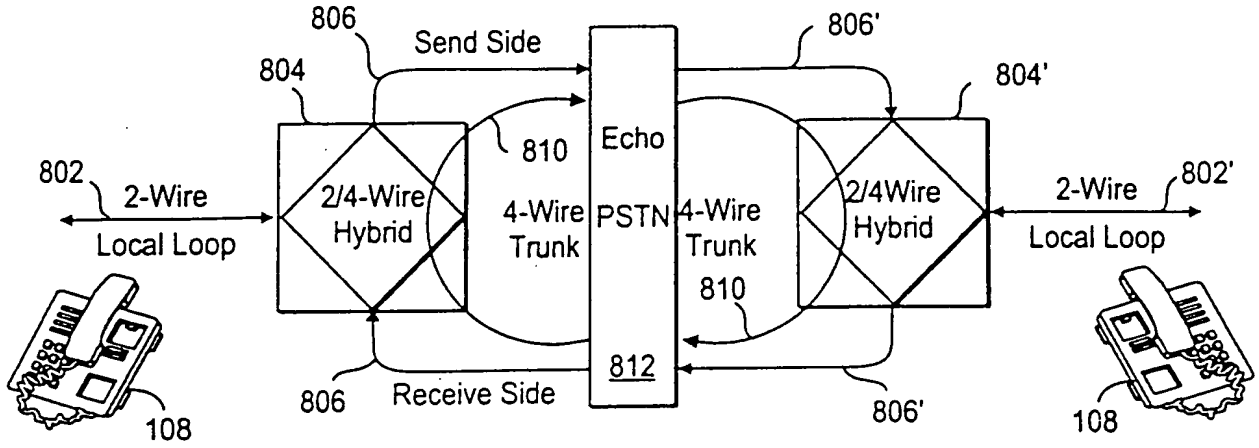


Fig. 8
(Prior Art)

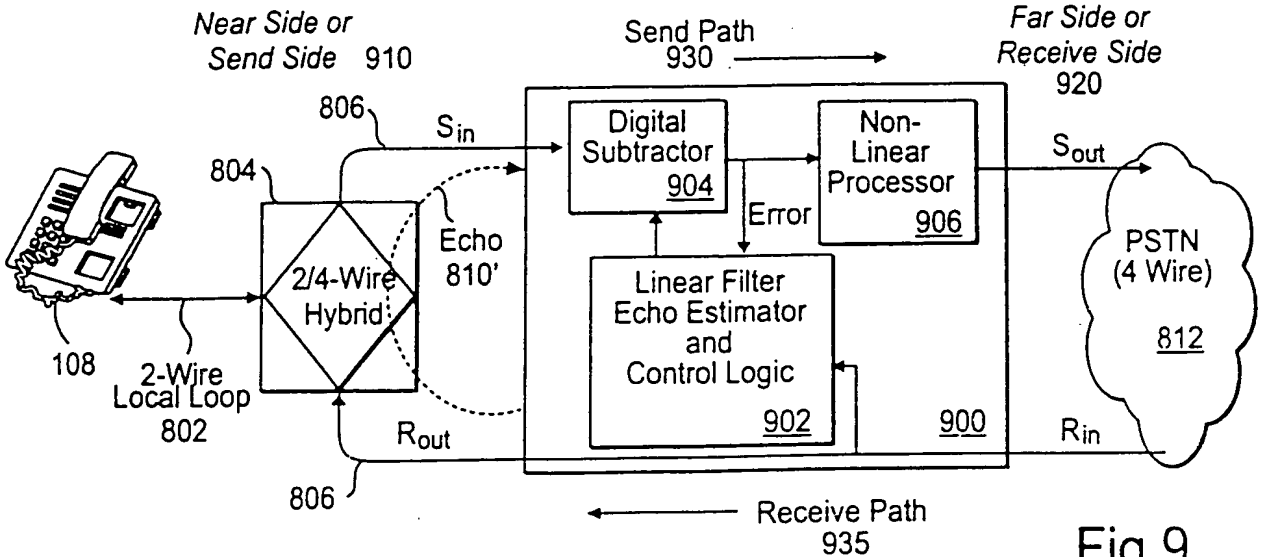


Fig.9
(Prior Art)

21/37

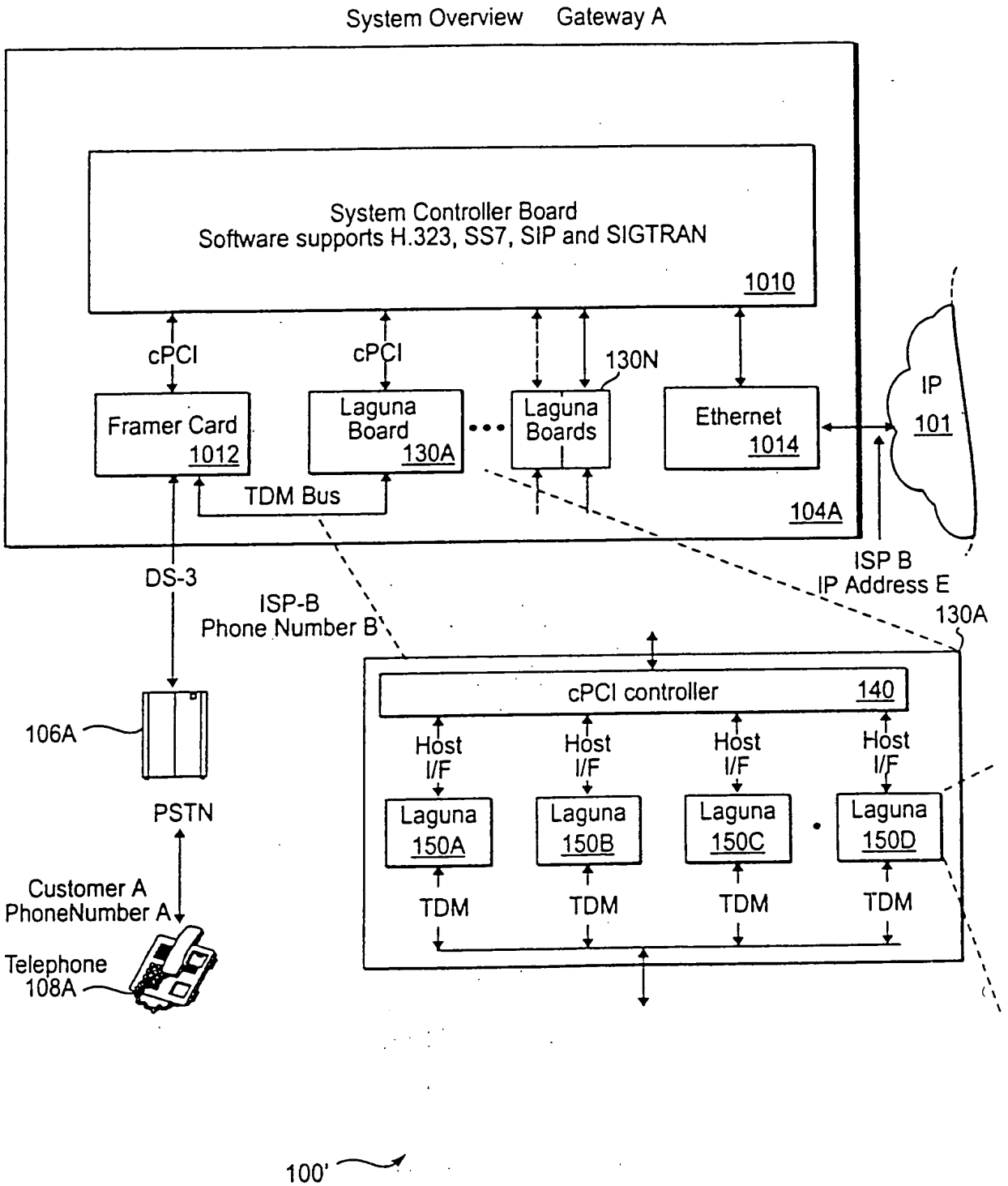

System Overview Gateway A
System Controller Board
Software supports H.323, SS7, SIP and SIGTRAN
1010
cPCI
cPCI
130N
Framer Card
1012
Laguna
Board
130A
Laguna
Boards
Ethernet
1014
IP
101
TDM Bus
104A
DS-3
ISP-B
Phone Number B
ISP B
IP Address E
130A
cPCI controller
140
Host
I/F
Host
I/F
Host
I/F
Host
I/F
Laguna
150A
Laguna
150B
Laguna
150C
Laguna
150D
TDM
TDM
TDM
TDM
106A
PSTN
Customer A
PhoneNumber A
Telephone
108A
100'

Fig.10(1)

22/37

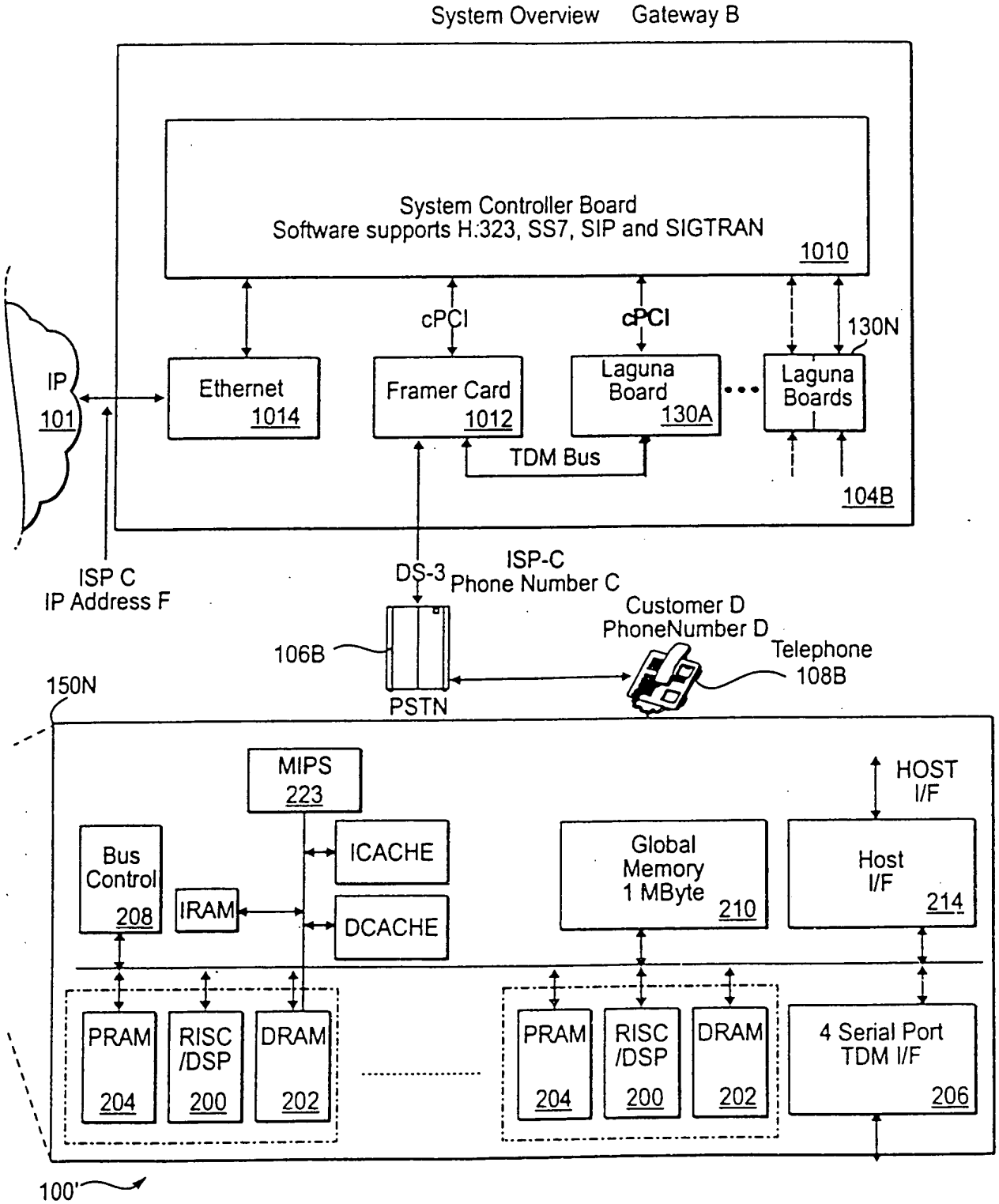

System Overview Gateway B
System Controller Board
Software supports H.323, SS7, SIP and SIGTRAN
1010
cPCI
cPCI
130N
IP
101
Ethernet
1014
Framer Card
1012
Laguna
Board
130A
Laguna
Boards
TDM Bus
104B
ISP C
IP Address F
DS-3
ISP-C
Phone Number C
Customer D
PhoneNumber D
106B
Telephone
108B
PSTN
150N
MIPS
223
HOST
I/F
Bus
Control
208
ICACHE
IRAM
DCACHE
Global
Memory
1 MByte
210
Host
I/F
214
PRAM
204
RISC
/DSP
200
DRAM
202
PRAM
204
RISC
/DSP
200
DRAM
202
4 Serial Port
TDM I/F
206
100'

Fig.10(2)

23/37

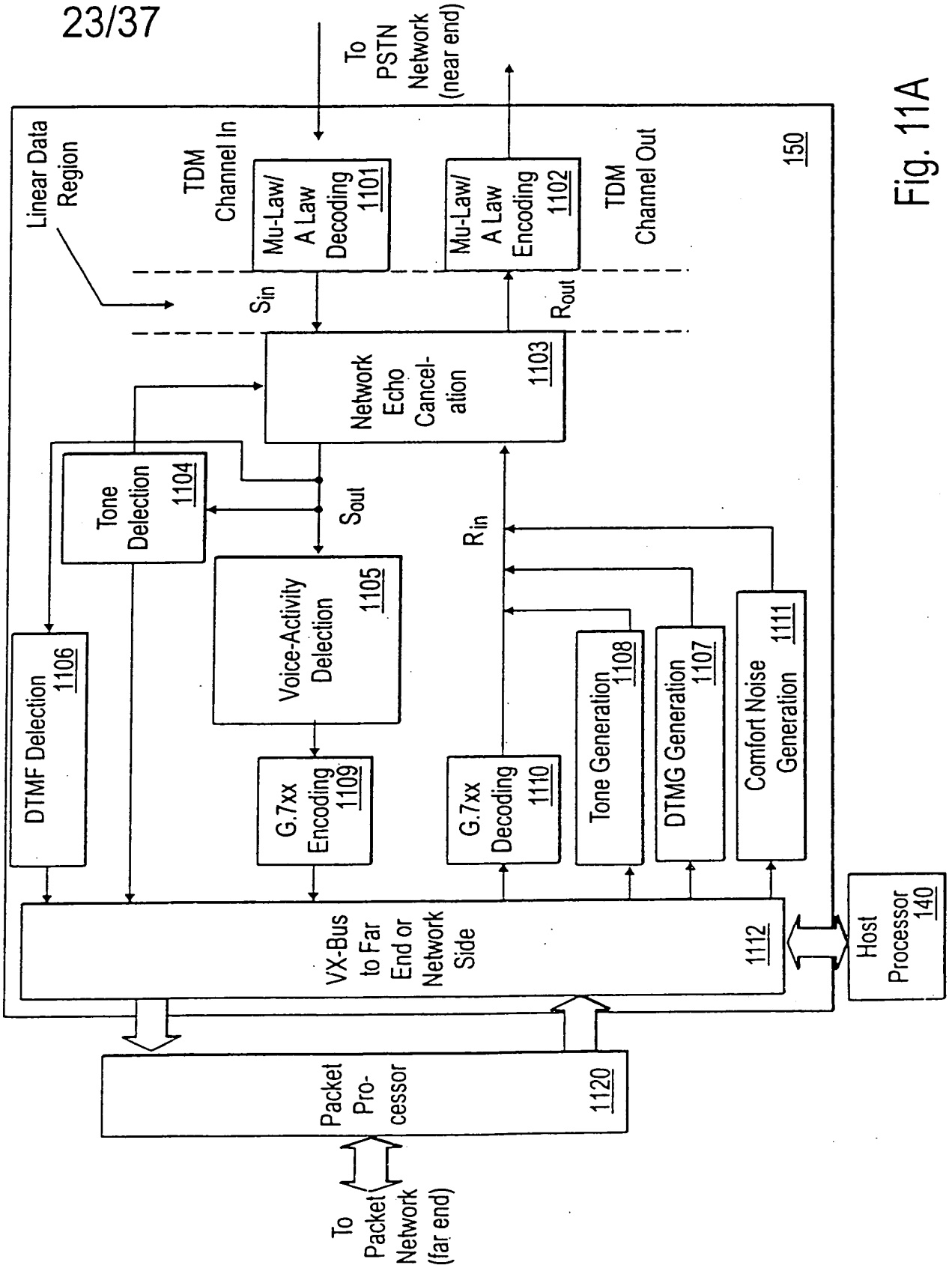

To PSTN Network (near end)
Linear Data Region
TDM Channel In
Mu-Law/ A Law Decoding 1101
Mu-Law/ A Law Encoding 1102
TDM Channel Out
150
Sin
Rout
Network Echo Cancel-ation 1103
Tone Delection 1104
Sout
Rin
Voice-Activity Delection 1105
DTMF Delection 1106
G.7xx Encoding 1109
G.7xx Decoding 1110
Tone Generation 1108
DTMG Generation 1107
Comfort Noise Generation 1111
VX-Bus to Far End or Network Side
1112
Host Processor 140
Packet Pro-cessor
1120
To Packet Network (far end)

Fig. 11A

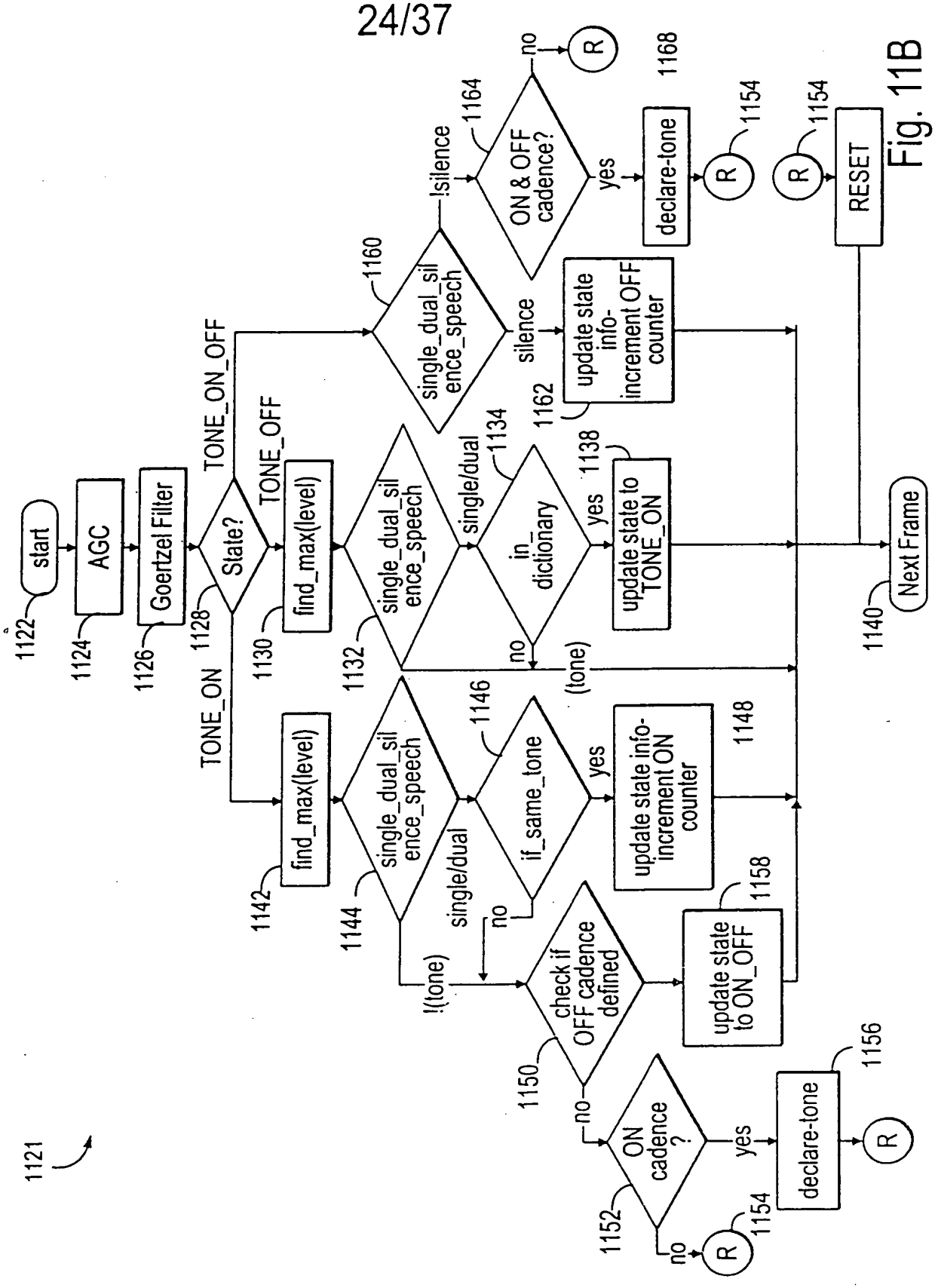

1121
start
1122
AGC
1124
Goertzel Filter
1126
State?
1128
TONE_ON
TONE_ON_OFF
TONE_OFF
find_max(level)
1130
single_dual_sil ence_speech
1132
single/dual
!(tone)
(tone)
in_ dictionary
1134
yes
no
update state to TONE_ON
1138
single_dual_sil ence_speech
1160
!silence
silence
ON & OFF cadence?
1164
yes
no
update state info- increment OFF counter
1162
declare-tone
1168
R
1154
RESET
Next Frame
1140
find_max(level)
1142
single_dual_sil ence_speech
1144
single/dual
!(tone)
if_same_tone
1146
yes
no
update state info- increment ON counter
1148
check if OFF cadence defined
1150
yes
no
update state to ON_OFF
1158
ON cadence ?
1152
yes
no
declare-tone
1156

Fig. 11B

25/37

Exemplary Filter Coefficients for Goertze Filter

frequency	cos(2* pi*f1/fs)	frequency index
350	31536	0
400	31163	1
425	30958	2
440	30829	3
480	30465	4
540	29863	5
600	29195	6
620	28958	7
660	28462	8
697	27978	9
700	27938	10
770	26955	11
780	26808	12
852	25700	13
900	24916	14
941	24218	15
1020	22802	16
1100	21280	17
1140	20487	18
1209	19072	19
1300	17120	20
1336	16324	21
1380	15332	22
1477	13084	23
1500	12539	24
1620	9634	25
1633	9314	26
1700	1649	27
1740	6644	28
1860	3595	29
1980	514	30
2040	-1029	31
2100	-2570	32
2280	-7147	33
2400	-10125	34
2600	-14875	35
3825	-32457	36

Fig. 11C

Exemplary Call Progress Tones

Frequency1	Frequency2	Call ProgressTone
350	440	ANSI T1.401 dial tone
425	0	Q.35 Dial Tone
440	480	ANSI T1.401 audiable ringing
480	620	ANSI T1.401 line busy tone
480	620	ANSI T1.401 Recorder
400	0	Audiable ringing
440	0	Dial Tone
440	0	ANSI T1.401Fast Busy Tone
440	0	Busy Tone

Fig. 11D

26/37

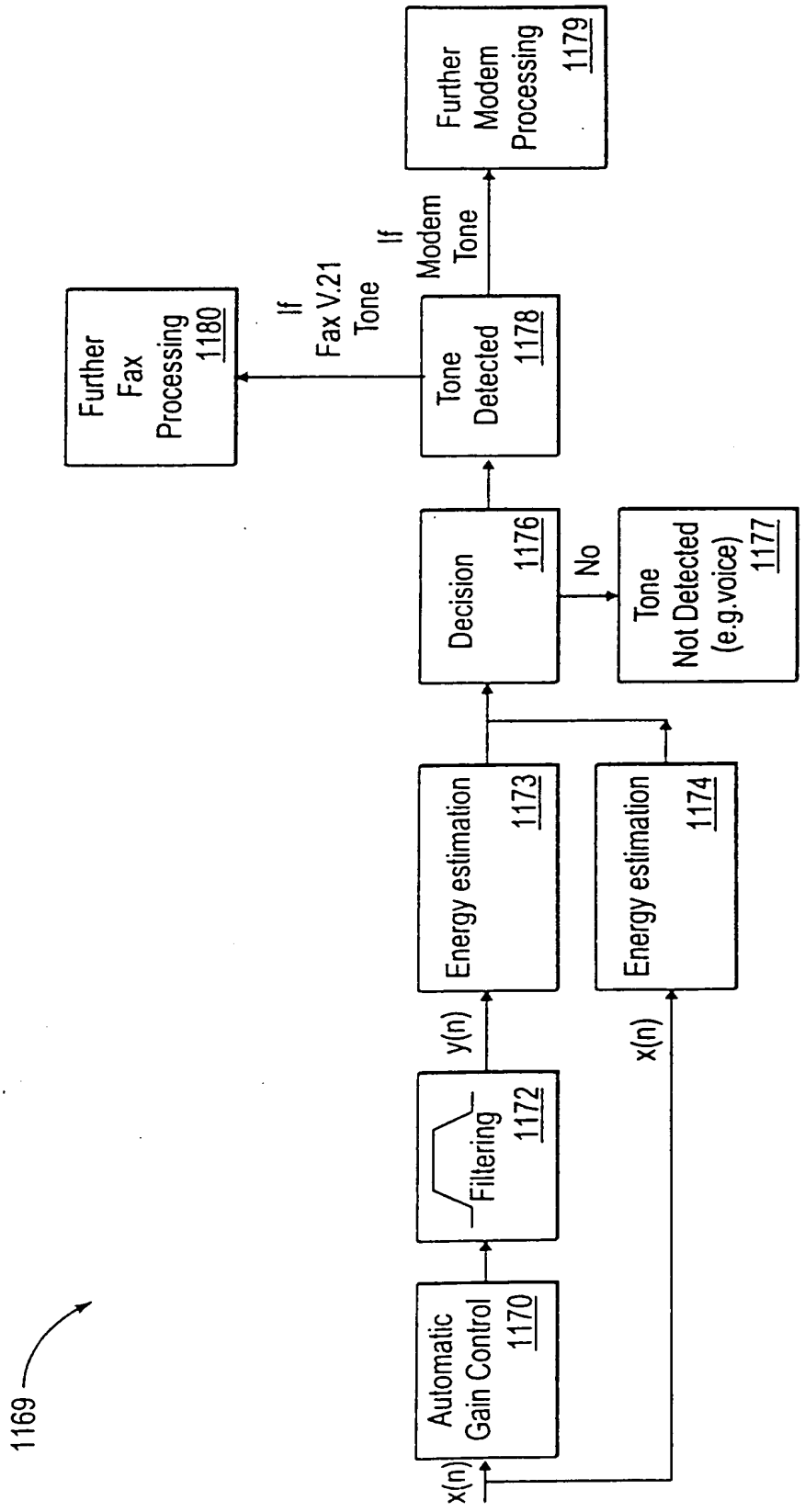

1169
Automatic Gain Control 1170
Filtering 1172
y(n)
x(n)
Energy estimation 1173
Energy estimation 1174
Decision 1176
No
Tone Not Detected (e.g.voice) 1177
Tone Detected 1178
If Fax V.21 Tone
If Modem Tone
Further Fax Processing 1180
Further Modem Processing 1179

Fig. 11E

27/37

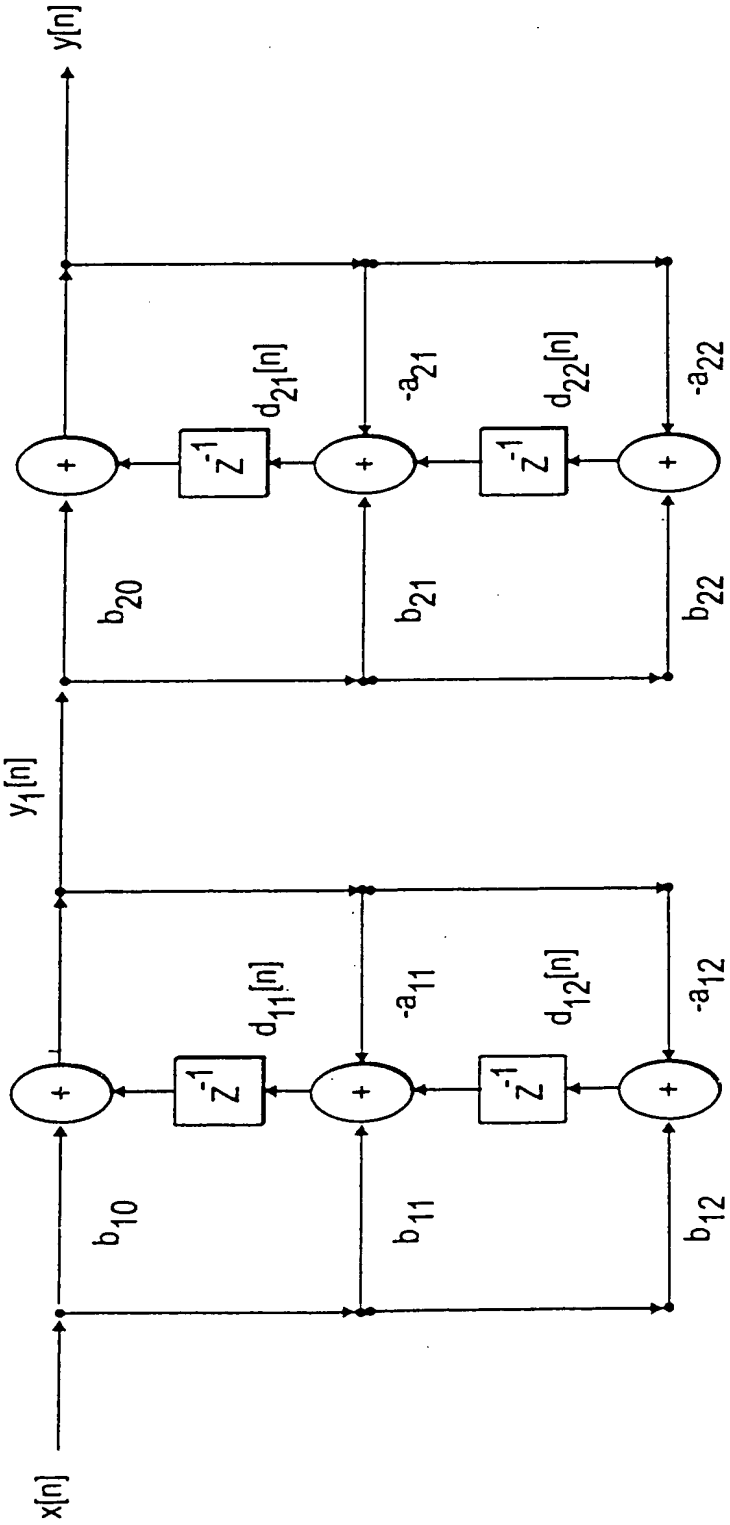
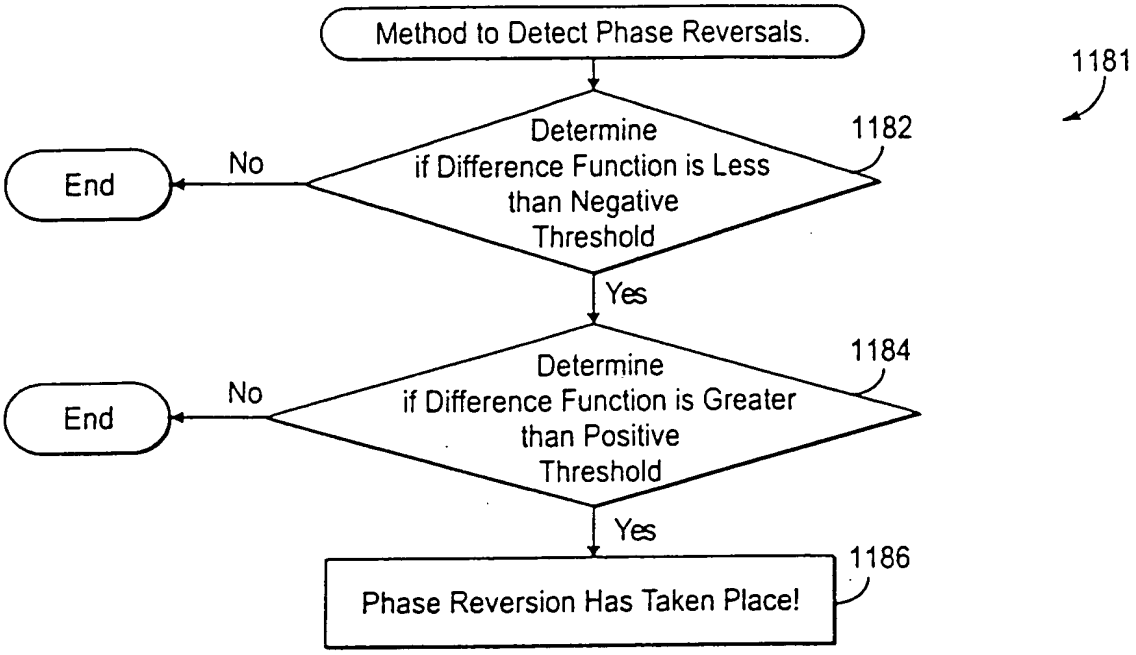

y[n]
$y_1[n]$
x[n]
z^{-1}
$d_{21}[n]$
$d_{22}[n]$
$d_{11}[n]$
$d_{12}[n]$
$-a_{21}$
$-a_{22}$
$-a_{11}$
$-a_{12}$
b_{20}
b_{21}
b_{22}
b_{10}
b_{11}
b_{12}

Fig. 11F

28/37


Method to Detect Phase Reversals.
1181
Determine if Difference Function is Less than Negative Threshold
1182
No
End
Yes
Determine if Difference Function is Greater than Positive Threshold
1184
No
End
Yes
Phase Reversion Has Taken Place!
1186

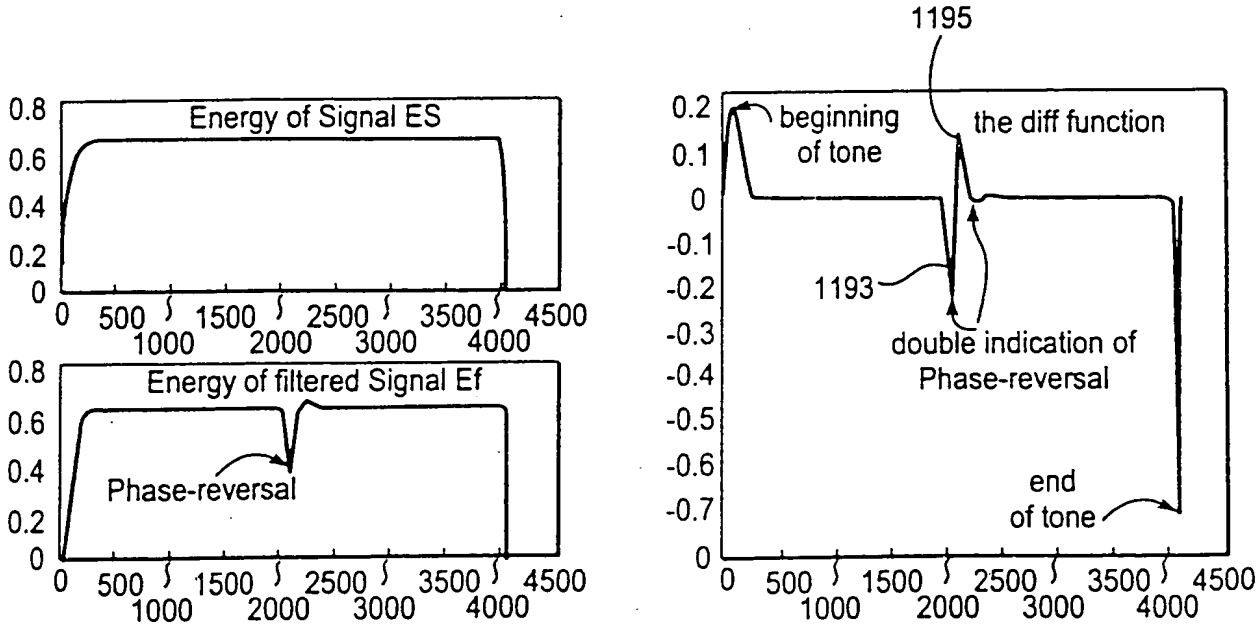

Energy of Signal ES
Energy of filtered Signal Ef
Phase-reversal
1195
beginning of tone
the diff function
1193
double indication of Phase-reversal
end of tone

Fig. 11G

29/37

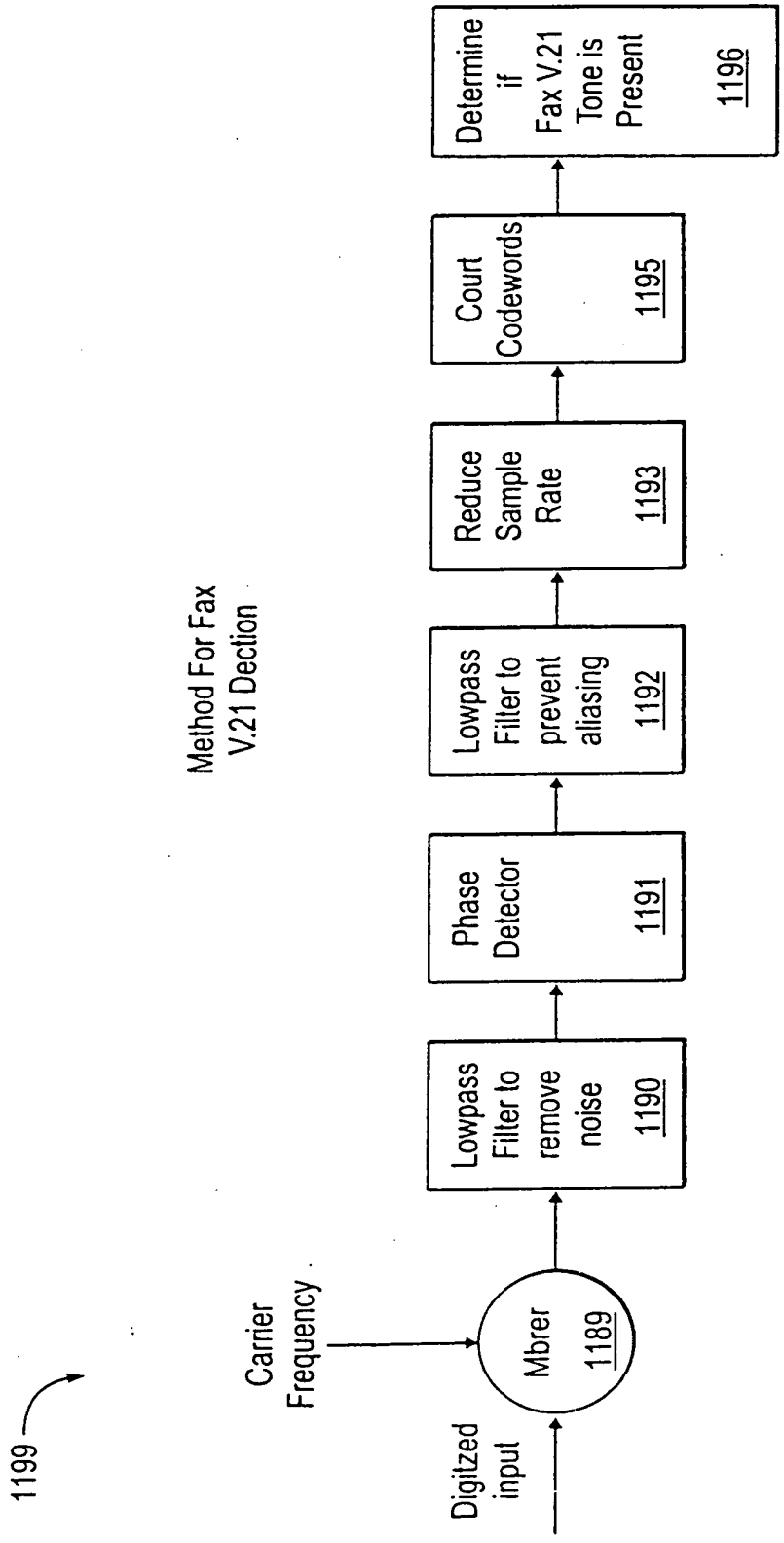

Method For Fax V.21 Dection
1199
Digitzed input
Carrier Frequency
Mbrer 1189
Lowpass Filter to remove noise 1190
Phase Detector 1191
Lowpass Filter to prevent aliasing 1192
Reduce Sample Rate 1193
Court Codewords 1195
Determine if Fax V.21 Tone is Present 1196

Fig. 11H

30/37

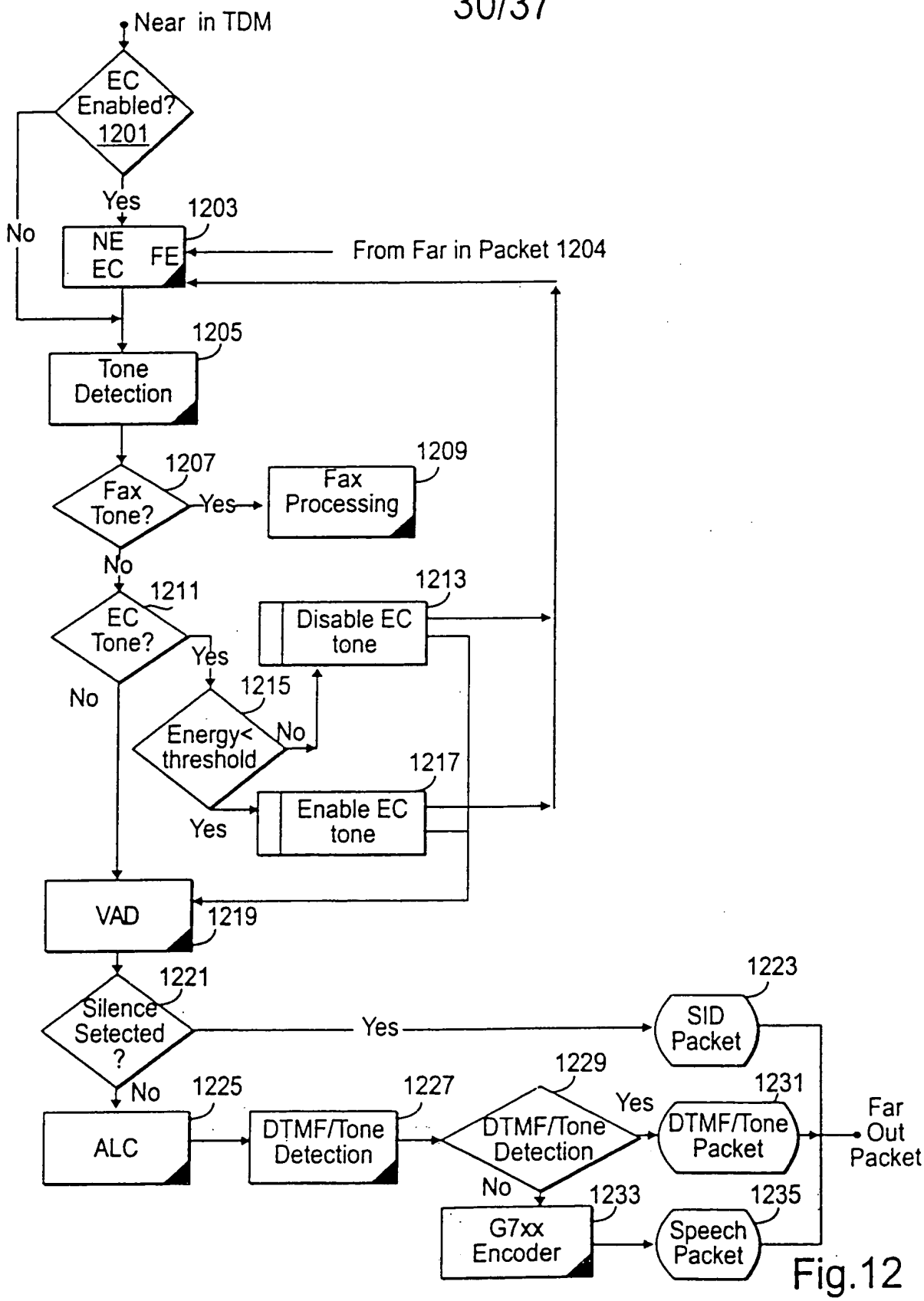

Near in TDM
EC Enabled? 1201
Yes
No
1203
NE EC FE
From Far in Packet 1204
1205
Tone Detection
1207
Fax Tone?
Yes
1209
Fax Processing
No
1211
EC Tone?
Yes
No
1213
Disable EC tone
1215
Energy< threshold
No
Yes
1217
Enable EC tone
VAD
1219
1221
Silence Setected ?
Yes
1223
SID Packet
No
1225
ALC
1227
DTMF/Tone Detection
1229
DTMF/Tone Detection
Yes
1231
DTMF/Tone Packet
Far Out Packet
No
1233
G7xx Encoder
1235
Speech Packet

Fig.12

31/37

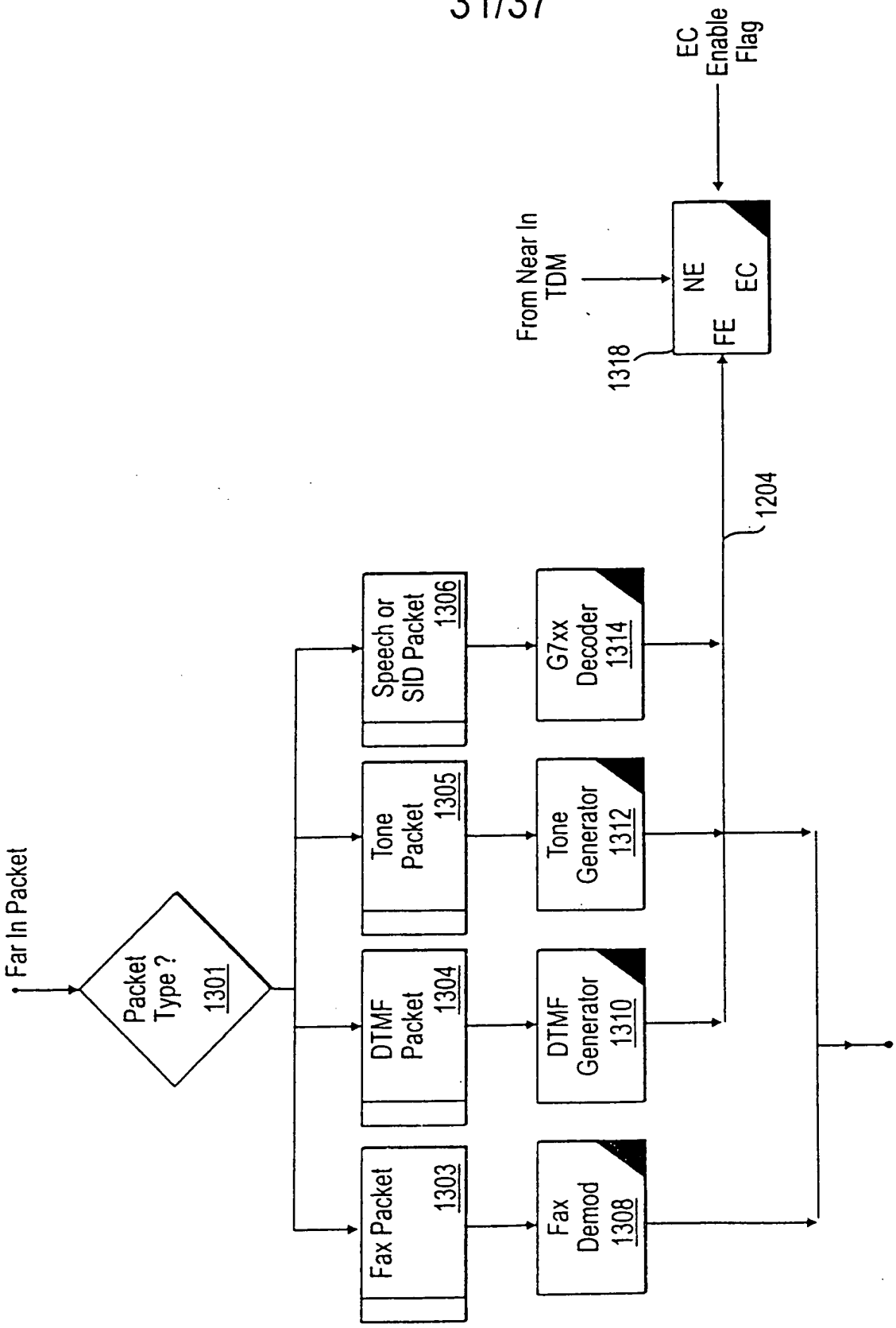

Far In Packet
Packet Type ? 1301
Fax Packet 1303
DTMF Packet 1304
Tone Packet 1305
Speech or SID Packet 1306
Fax Demod 1308
DTMF Generator 1310
Tone Generator 1312
G7xx Decoder 1314
1204
1318
NE FE EC
From Near In TDM
EC Enable Flag
Near Out TDM

Fig. 13

32/37

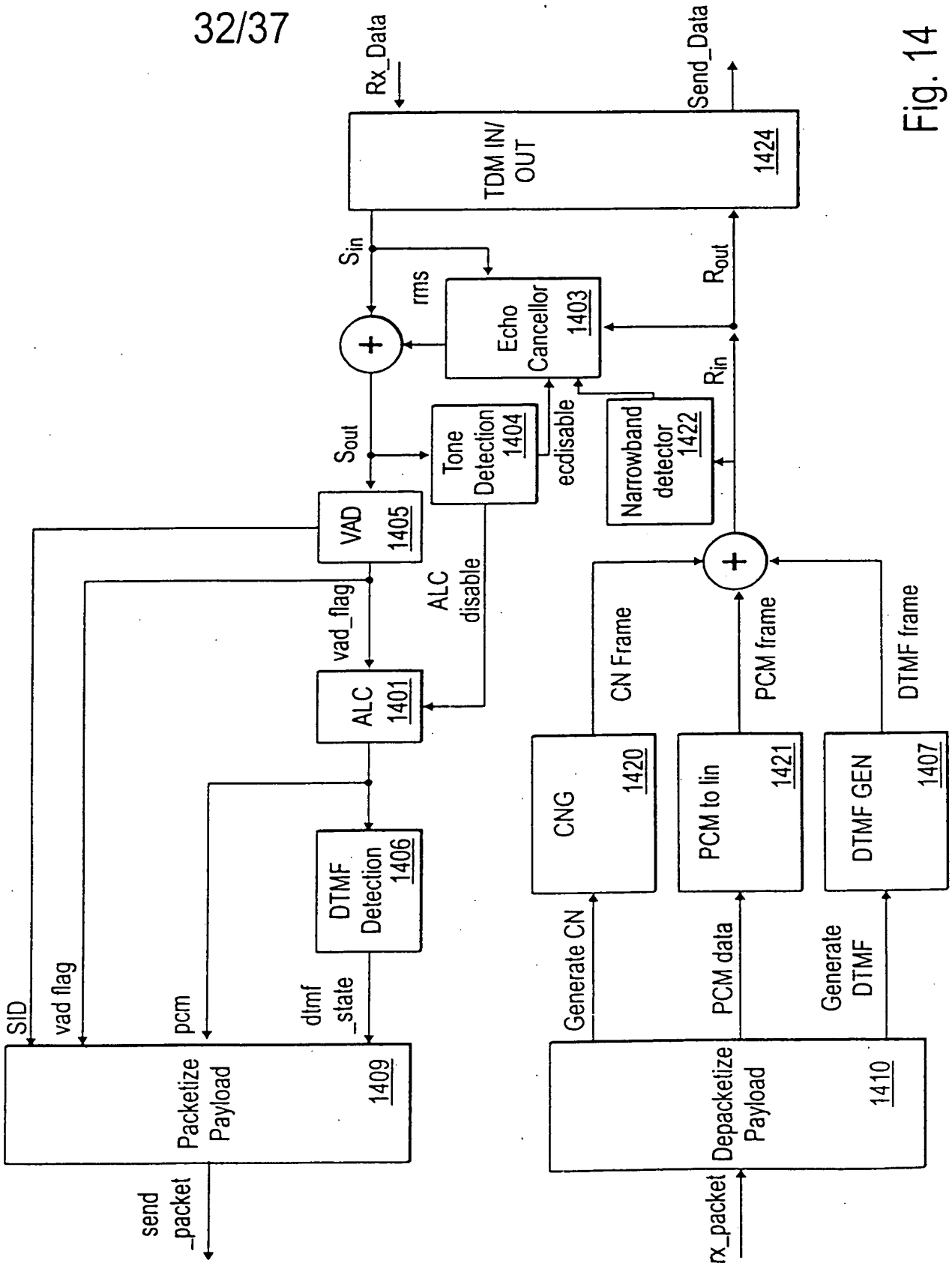

Rx_Data
Send_Data
TDM IN/ OUT
1424
Sin
rms
Echo Cancellor 1403
Rout
Rin
Sout
Tone Detection 1404
ecdisable
Narrowband detector 1422
VAD 1405
ALC disable
vad_flag
ALC 1401
CN Frame
PCM frame
DTMF frame
CNG 1420
PCM to lin 1421
DTMF GEN 1407
DTMF Detection 1406
SID
vad flag
pcm
dtmf _state
Generate CN
PCM data
Generate DTMF
Packetize Payload 1409
Depacketize Payload 1410
send _packet
rx_packet

Fig. 14

33/37

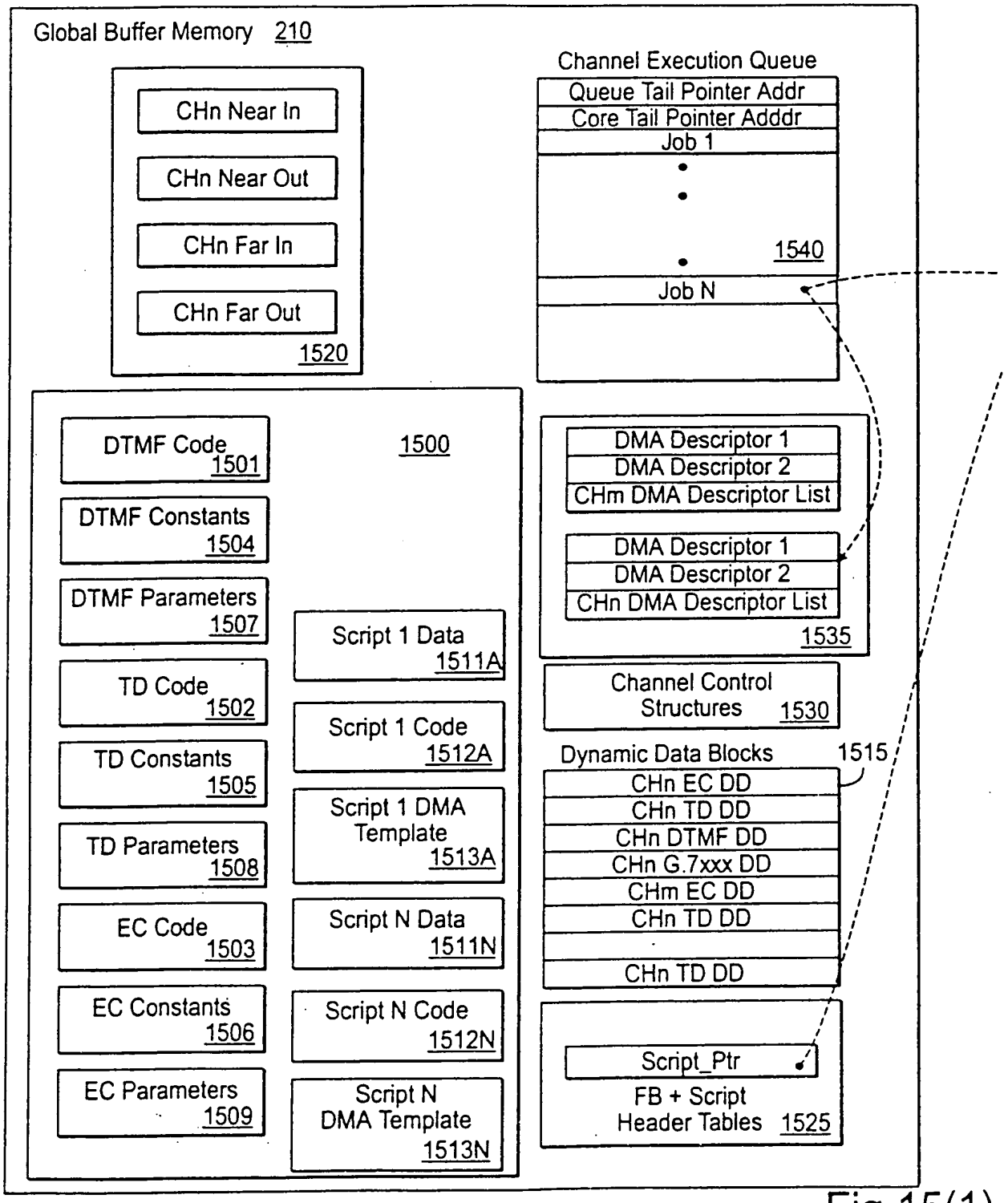

Global Buffer Memory 210
CHn Near In
CHn Near Out
CHn Far In
CHn Far Out
1520
Channel Execution Queue
Queue Tail Pointer Addr
Core Tail Pointer Adddr
Job 1
1540
Job N
1500
DTMF Code 1501
DTMF Constants 1504
DTMF Parameters 1507
TD Code 1502
TD Constants 1505
TD Parameters 1508
EC Code 1503
EC Constants 1506
EC Parameters 1509
Script 1 Data 1511A
Script 1 Code 1512A
Script 1 DMA Template 1513A
Script N Data 1511N
Script N Code 1512N
Script N DMA Template 1513N
DMA Descriptor 1
DMA Descriptor 2
CHm DMA Descriptor List
DMA Descriptor 1
DMA Descriptor 2
CHn DMA Descriptor List
1535
Channel Control Structures 1530
Dynamic Data Blocks 1515
CHn EC DD
CHn TD DD
CHn DTMF DD
CHn G.7xxx DD
CHm EC DD
CHn TD DD
CHn TD DD
Script_Ptr
FB + Script Header Tables 1525

Fig.15(1)

34/37

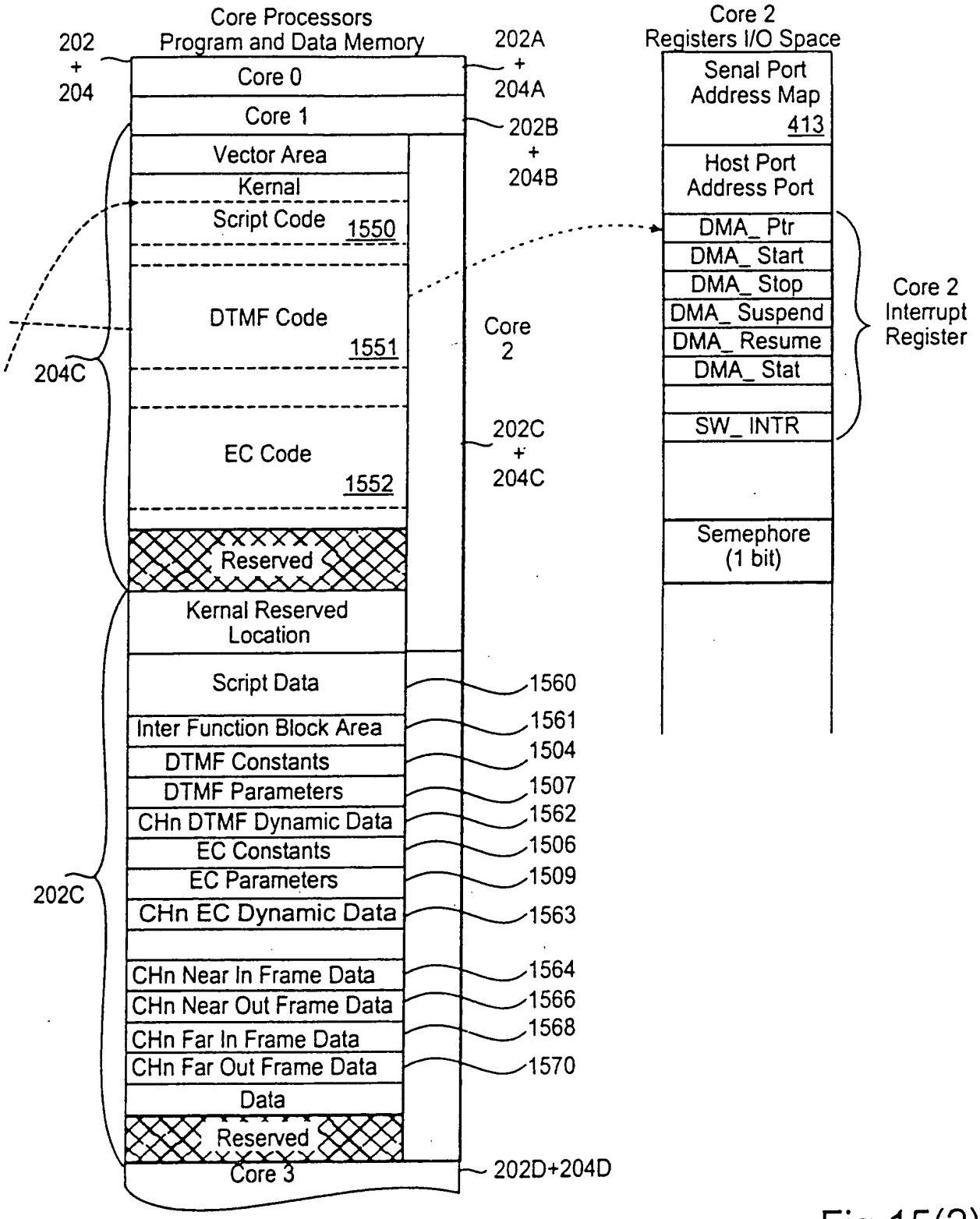

Core Processors
Program and Data Memory
202
+
204
Core 0
Core 1
Vector Area
Kernal
Script Code 1550
DTMF Code 1551
EC Code 1552
Reserved
Kernal Reserved Location
Script Data
Inter Function Block Area
DTMF Constants
DTMF Parameters
CHn DTMF Dynamic Data
EC Constants
EC Parameters
CHn EC Dynamic Data
CHn Near In Frame Data
CHn Near Out Frame Data
CHn Far In Frame Data
CHn Far Out Frame Data
Data
Reserved
Core 3
202A
+
204A
202B
+
204B
Core 2
202C
+
204C
202D+204D
204C
202C
1560
1561
1504
1507
1562
1506
1509
1563
1564
1566
1568
1570
Core 2
Registers I/O Space
Senal Port Address Map 413
Host Port Address Port
DMA_ Ptr
DMA_ Start
DMA_ Stop
DMA_ Suspend
DMA_ Resume
DMA_ Stat
SW_ INTR
Core 2 Interrupt Register
Semephore (1 bit)

Fig.15(2)

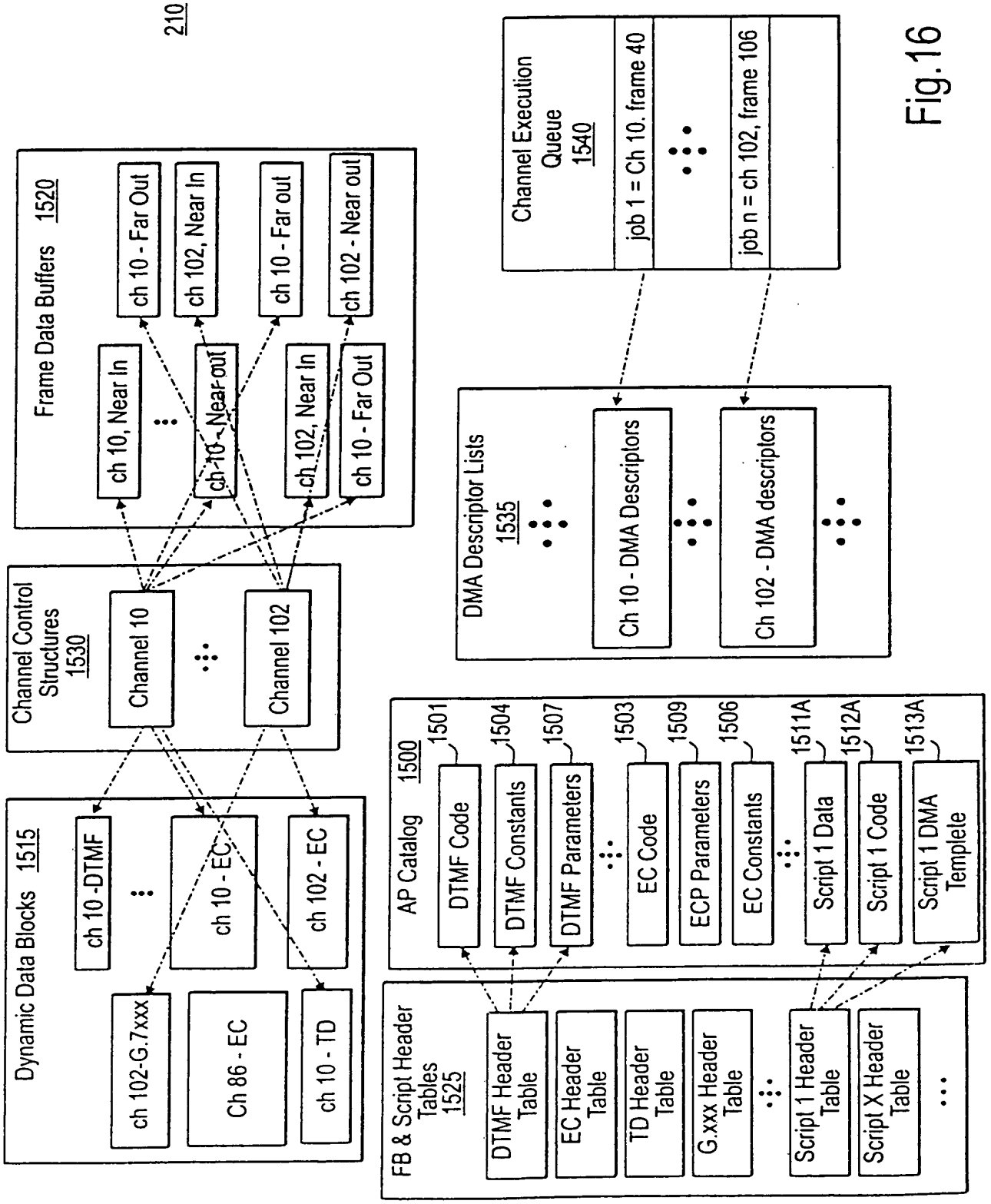

210
Frame Data Buffers 1520
ch 10, Near In
ch 10 - Near out
ch 10 - Far Out
ch 102, Near In
ch 102 - Near out
ch 10 - Far out
ch 102, Near In
ch 10 - Far Out
Channel Execution Queue 1540
job 1 = Ch 10. frame 40
job n = ch 102, frame 106
DMA Descriptor Lists 1535
Ch 10 - DMA Descriptors
Ch 102 - DMA descriptors
Channel Control Structures 1530
Channel 10
Channel 102
Dynamic Data Blocks 1515
ch 10 -DTMF
ch 10 - EC
ch 102 - EC
ch 102-G.7xxx
Ch 86 - EC
ch 10 - TD
AP Catalog 1500
1501
DTMF Code
1504
DTMF Constants
1507
DTMF Parameters
1503
EC Code
1509
ECP Parameters
1506
EC Constants
1511A
Script 1 Data
1512A
Script 1 Code
1513A
Script 1 DMA Templete
FB & Script Header Tables 1525
DTMF Header Table
EC Header Table
TD Header Table
G.xxx Header Table
Script 1 Header Table
Script X Header Table

Fig.16

36/37

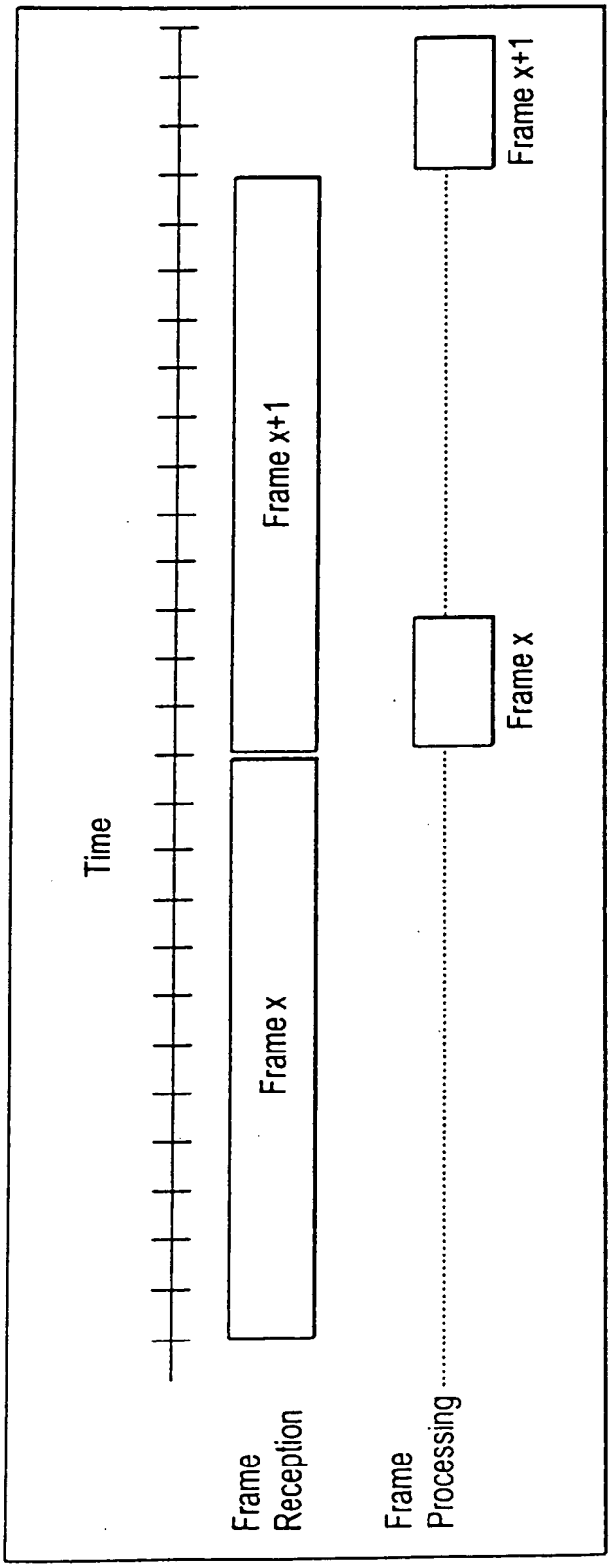

Time
Frame x
Frame x+1
Frame x
Frame x+1
Frame Reception
Frame Processing

Fig. 17

37/37

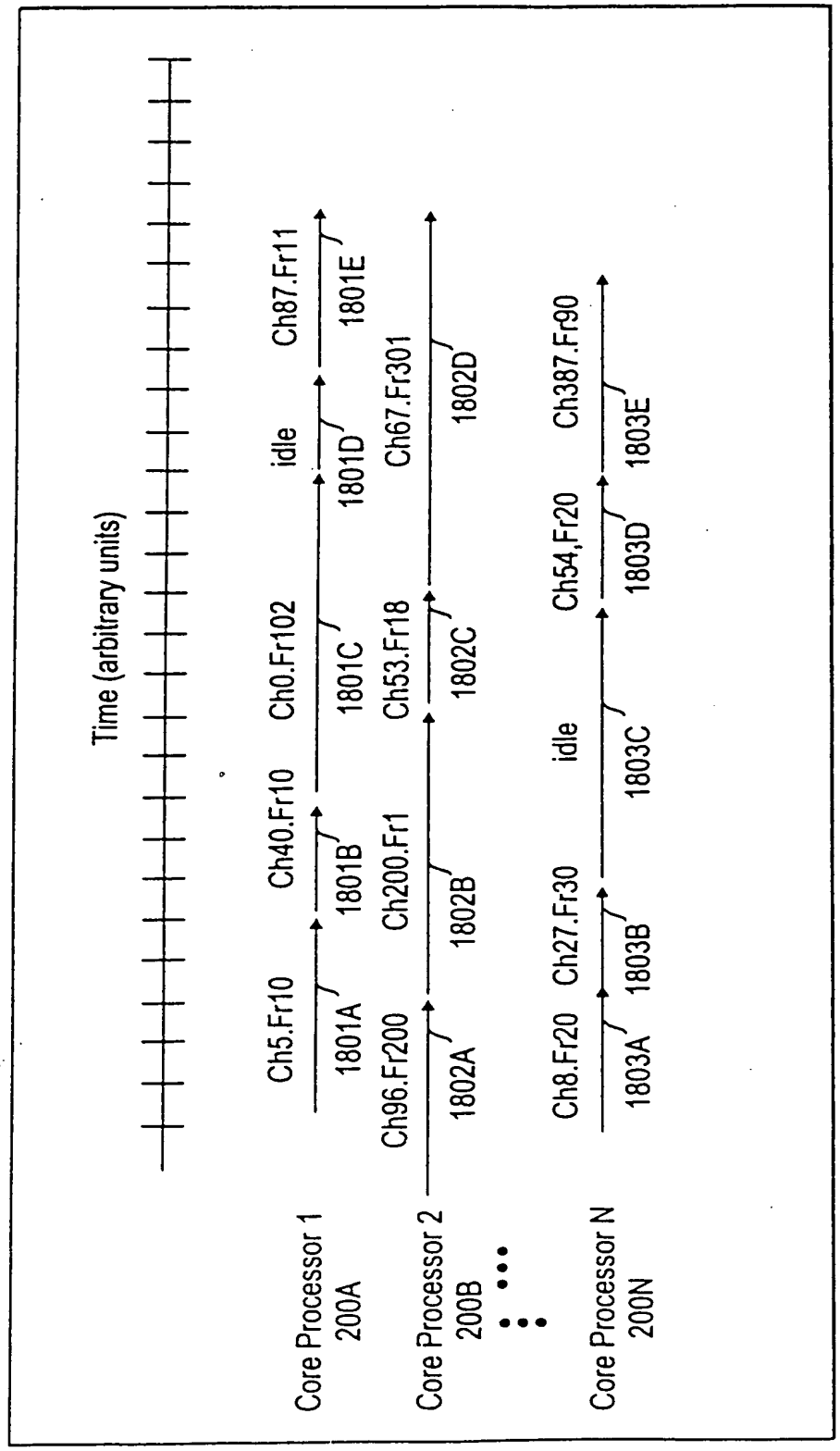

Time (arbitrary units)
Core Processor 1
200A
Core Processor 2
200B
Core Processor N
200N
Ch5.Fr10
Ch40.Fr10
Ch0.Fr102
idle
Ch87.Fr11
1801A
1801B
1801C
1801D
1801E
Ch96.Fr200
Ch200.Fr1
Ch53.Fr18
Ch67.Fr301
1802A
1802B
1802C
1802D
Ch8.Fr20
Ch27.Fr30
idle
Ch54,Fr20
Ch387.Fr90
1803A
1803B
1803C
1803D
1803E

Fig. 18